

DISEÑO Y CONSTRUCCIÓN DE VALLA ROBÓTICA INTERACTIVA PARA
PUBLICIDAD. MODULO 2

AUGUSTO ARENAS VELASQUEZ
JOSÉ FRANCISCO RODRÍGUEZ TRILLOS

UNIVERSIDAD AUTÓNOMA DE OCCIDENTE
FACULTAD DE INGENIERIA
DEPARTAMENTO DE AUTOMATICA Y ELECTRONICA
PROGRAMA INGENIERÍA MECATRÓNICA
SANTIAGO DE CALI
2008

DISEÑO Y CONSTRUCCIÓN DE VALLA ROBÓTICA INTERACTIVA PARA
PUBLICIDAD. MODULO 2

AUGUSTO ARENAS VELASQUEZ
JOSÉ FRANCISCO RODRÍGUEZ TRILLOS

Pasantía para optar al título de Ingeniero Mecatrónico

Director
DRAGO DUSSICH
Ingeniero Mecatrónico

UNIVERSIDAD AUTÓNOMA DE OCCIDENTE
FACULTAD DE INGENIERIA
DEPARTAMENTO DE AUTOMATICA Y ELECTRONICA
PROGRAMA INGENIERÍA MECATRÓNICA
SANTIAGO DE CALI
2008

Nota de aceptación

Aprobado por el comité de grado en cumplimiento de los requisitos exigidos por la Universidad Autónoma de Occidente para optar el título de Ingeniero Mecatrónico.

Ing. DRAGO DUSSICH

Director

Santiago de Cali, 11 de Octubre de 2008

CONTENIDO

	Pág.
RESUMEN	10
INTRODUCCIÓN	12
1. OBJETIVOS	13
1.1 GENERAL	13
1.2 ESPECIFICOS	13
2. ANTECEDENTES	14
2.1 ESQUEMA DE PRODUCTO	14
2.2 ELEMENTOS FISICOS Y FUNCIONALES	15
2.3 DISTRIBUCIÓN GEOMÉTRICA DE LOS COMPONENTES	16
2.4 ARQUITECTURA DEL SISTEMA ELECTRONICO	18
2.4.1 Circuito principal de control	19
2.4.2 Circuito microcontrolador	19
2.5 SISTEMA DE MOVIMIENTOS	20
2.5.1 Movimiento lineal	20
2.5.2 Movimiento rotacional	21
2.5.3 Movimientos especiales	22
2.6 DIAGRAMA DE FLUJO	23
2.7 ESPECIFICACIONES FINALES	24
3. ESPECIFICACIONES DE SOFTWARE	29
3.1 SISTEMA OPERATIVO	29
3.2 FUNCIONAMIENTO GENERAL DEL SOFTWARE	30

	Pág.
3.3 SISTEMAS DE COMUNICACION	33
3.3.1 Puerto serial	33
3.3.1.1 Descripción general	34
3.3.1.2 Especificación de uso	35
3.3.2 Puerto paralelo	38
3.3.2.1 Descripción general	38
3.3.2.2 Especificación de uso	40
3.4 SISTEMA REPRODUCTOR DE AUDIO DIGITAL	40
3.4.1 Formato MP3	40
3.4.2 Software de reproducción MPG321	40
3.5 INTERFAZ GRÁFICA DE USUARIO	41
4. ESPECIFICACIONES DE HARDWARE	43
4.1 DESCRIPCIÓN DETALLADA DEL SISTEMA DE CONTROL	43
4.2 BOARDS O CIRCUITOS IMPRESOS	43
4.2.1 Board de comunicaciones (Board 1)	44
4.2.2 Board de conexiones para servomotores (Board 2)	48
4.2.3 Board de potencia (Board 3)	50
5. DISEÑO Y ARQUITECTURA	55
5.1 DIAGRAMA DE CLASES	55
6. CONCLUSIONES	57
BIBLIOGRAFÍA	59
ANEXOS	60

LISTA DE TABLAS

	Pág.
Tabla 1. Especificaciones finales	25
Tabla 2. Tabla de convenciones del esquema funcional	26
Tabla 3. Disposición de pines en el puerto serial	34
Tabla 4. Estructura Termios	36
Tabla 5. Opciones de control	37
Tabla 6. Tabla de especificaciones de la BOARD 1	47
Tabla 7. Tabla de especificaciones de la BOARD 2	50
Tabla 8. Tabla de especificaciones de la BOARD 3	54

LISTA DE FIGURAS

	Pág.
Figura 1. Esquema de producto	14
Figura 2. Elementos físicos y funcionales	15
Figura 3. Distribución geométrica	16
Figura 4. Distribución de componentes	17
Figura 5. Arquitectura del sistema electrónico	18
Figura 6. Diagrama esquemático electrónico	19
Figura 7. Movimiento lineal	20
Figura 8. Movimiento rotacional concéntrico	21
Figura 9. Movimiento rotacional excéntrico	21
Figura 10. Cilindro	22
Figura 11. Cubo	22
Figura 12. Diagrama de flujo	23
Figura 13. Animak	24
Figura 14. Esquema funcional	26
Figura 15. Esquema puerto serial	33
Figura 16. Esquema puerto paralelo	38
Figura 17. Interfaz de administración	42
Figura 18. Configuración de relojes	42
Figura 19. Interfaz gráfica de usuario	42
Figura 20. Board 1	45
Figura 21. Diagrama esquemático de la BOARD 1	46
Figura 22. Board 2	48

	Pág.
Figura 23. Diagrama esquemático de la BOARD 2	49
Figura 24. Board 3	52
Figura 25. Diagrama esquemático de la BOARD 3	53
Figura 26. Software, sistema de control de reproducción	55
Figura 27. Software, sistema de control general	56

LISTA DE ANEXOS

	Pág.
Anexo A. Documentación de LINUX	60
Anexo B. Cronograma proyecto ANIMAK	64

RESUMEN

En la búsqueda constante de formas de llegar al consumidor, las agencias de publicidad han agotado los medios actuales, dejando atrás la innovación y la capacidad de sorprender al espectador; y aunque estos medios siguen marcando la pauta mas alta en este campo, los grandes avances en la tecnología actual y el nacimiento en Colombia de carreras como la Ingeniería Mecatrónica, a dado nuevas ideas y esta abriendo nuevos horizontes a la publicidad moderna.

Productos informativos, interactivos, innovadores, de alta tecnología y versátiles son características indispensables para el desarrollo de dispositivos exitosos en el mercado publicitario, y son estas las que lo hacen perfecto para el desarrollo de un dispositivo mecatrónico. Podría parecer ciencia ficción, que actualmente exista un medio publicitario in-store que permita al consumidor, no solo conocer una marca o participar en una promoción, sino que también tener una experiencia visual y auditiva totalmente interactiva con el medio.

La información de doble vía o la capacidad de retro-alimentar al medio para que pueda adaptarse a los requerimientos del mercado, es una de las premisas mas importantes de este producto, permitiendo a la marca, recibir información relevante acerca que cuantos usuarios interactuan con el dispositivo o realizar encuestas directamente al publico que mira su publicidad.

Esto unido a la versatilidad del medio, la cual le permitirá, no solo estar en tiendas y centros comerciales, sino también en museos, parques temáticos, hoteles, restaurantes, aeropuertos, salas de cine, etc.

El desarrollo de este producto no solo aporta ideas y conceptos nuevos a la publicidad actual en Colombia, sino que pretende demostrar la gran habilidad del ingeniero mecatrónico para moverse entre diferentes disciplinas y prueba que su campo de acción esta en cualquier parte y depende básicamente de su intención de cambiar y transformar la forma en que se hacen las cosas; así como abrirle paso a la ingeniería mecatrónica en una de las industrias mas importantes a nivel nacional e internacional.

Este informe documenta el trabajo realizado durante el desarrollo de la pasantía en la empresa KAFKA LTDA. En este informe se detallan las etapas que se llevaron a cabo para lograr el diseño y la construcción de la VALLA ROBÓTICA INTERACTIVA

PARA PUBLICIDAD, que de ahora en adelante, para abreviar la llamaremos "ANIMAK".

Los objetivos propuestos por la empresa KAFKA LTDA para el desarrollo de ésta pasantía fueron acogidos por los pasantes de forma general, apoyándose mutuamente en todo momento, y también dedicándose individualmente, en los temas que mejor manejaba cada uno, pero siempre conservando la comunicación que era la clave durante el desarrollo de los diferentes sistemas y módulos de la valla ANIMAK.

INTRODUCCIÓN

Con el fin de romper los paradigmas de la publicidad moderna y crear medios que sorprendan a los consumidores; Kafka ha creado un sistema robótico de vallas interactivo, que permite a los usuarios, tener una experiencia multimedia completa y ofrecer al mismo tiempo a los anunciantes un mecanismo de marketing en tiempo real.

El desarrollo de este producto requiere de un robusto sistema de control que satisfaga las necesidades del mismo, ofrezca una interfaz amable, configurable y eficiente; y que permita a sus operarios, la constante actualización y mejora. Este proyecto desarrolla este sistema.

El software desarrollado maneja la información de la animación mecánica de la valla y la transmite a un microprocesador para que éste genere los movimientos en los actuadores. También es el encargado de manejar el audio, generar y proyectar la interfaz, capturar la interacción del usuario, administrar el sistema, reconocer dispositivos externos (Memorias USB, impresoras, etc.), generar los estudios de mercadeo y encuestas del producto; y también es el encargado de la puesta en marcha y posterior apagado de la valla.

Para desarrollar un sistema estándar y de gran flexibilidad, se ha optado por utilizar un sistema Linux como columna vertebral del proyecto, esto nos asegura una tranquilidad en cuanto a la adquisición de licencias de desarrollo de las herramientas utilizadas para la creación del sistema y nos permite utilizar la gran base de conocimiento sobre este sistema, que abunda en la Internet. Este sistema ha sido adaptado para los servicios que requiere la valla.

1. OBJETIVOS

1.1. GENERAL

- Diseñar y desarrollar el sistema de control para las vallas Animak de Kafka.

1.2. ESPECIFICOS

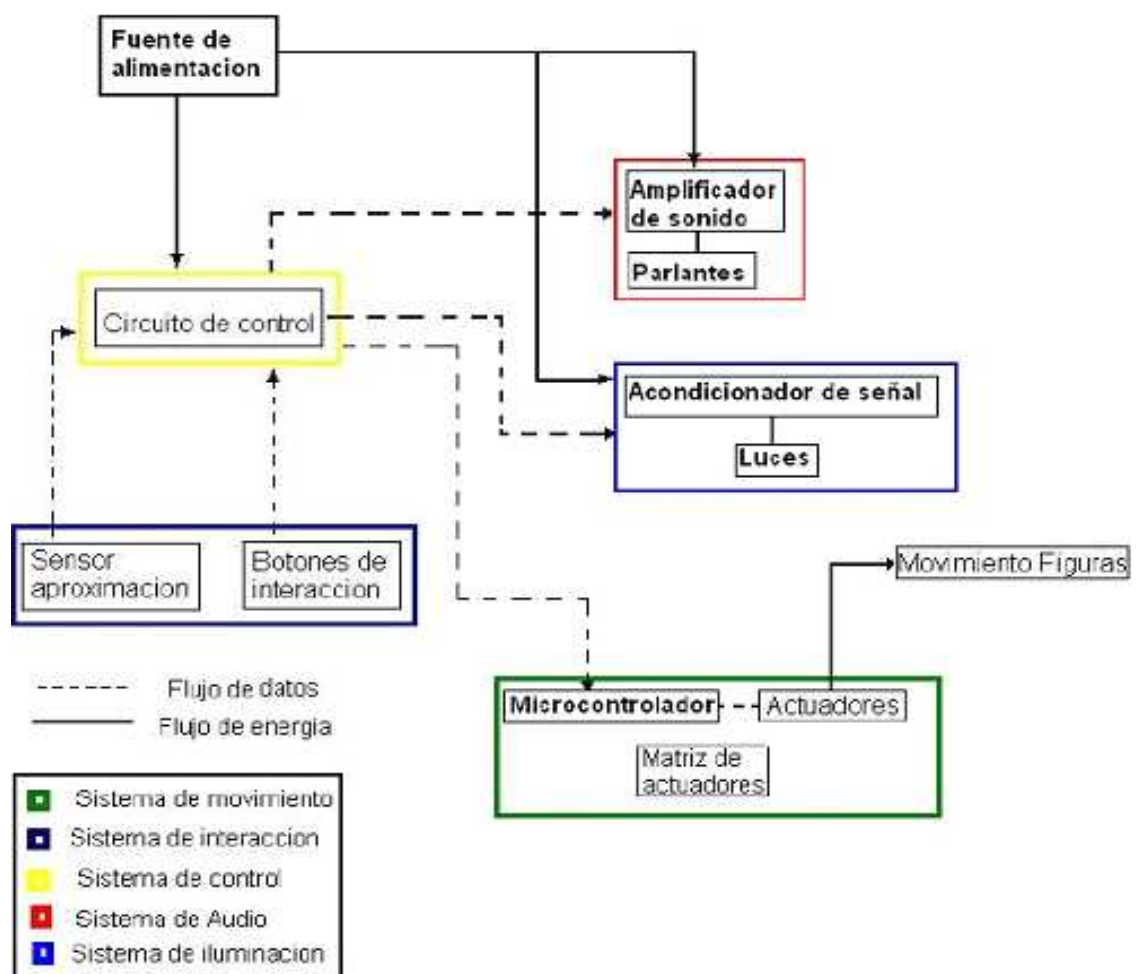
- Desarrollar e implementar el software de control para valla.
- Integrar el software del PC con el hardware del PIC .
- Diseñar e implementar el sistema de control de actuadores.
- Diseñar e implementar el sistema de potencia.
- Diseño de mecanismos.

2. ANTECEDENTES

Para contextualizar el desarrollo de este proyecto, se presentan a continuación, las principales características de la valla Animak de Kafka.

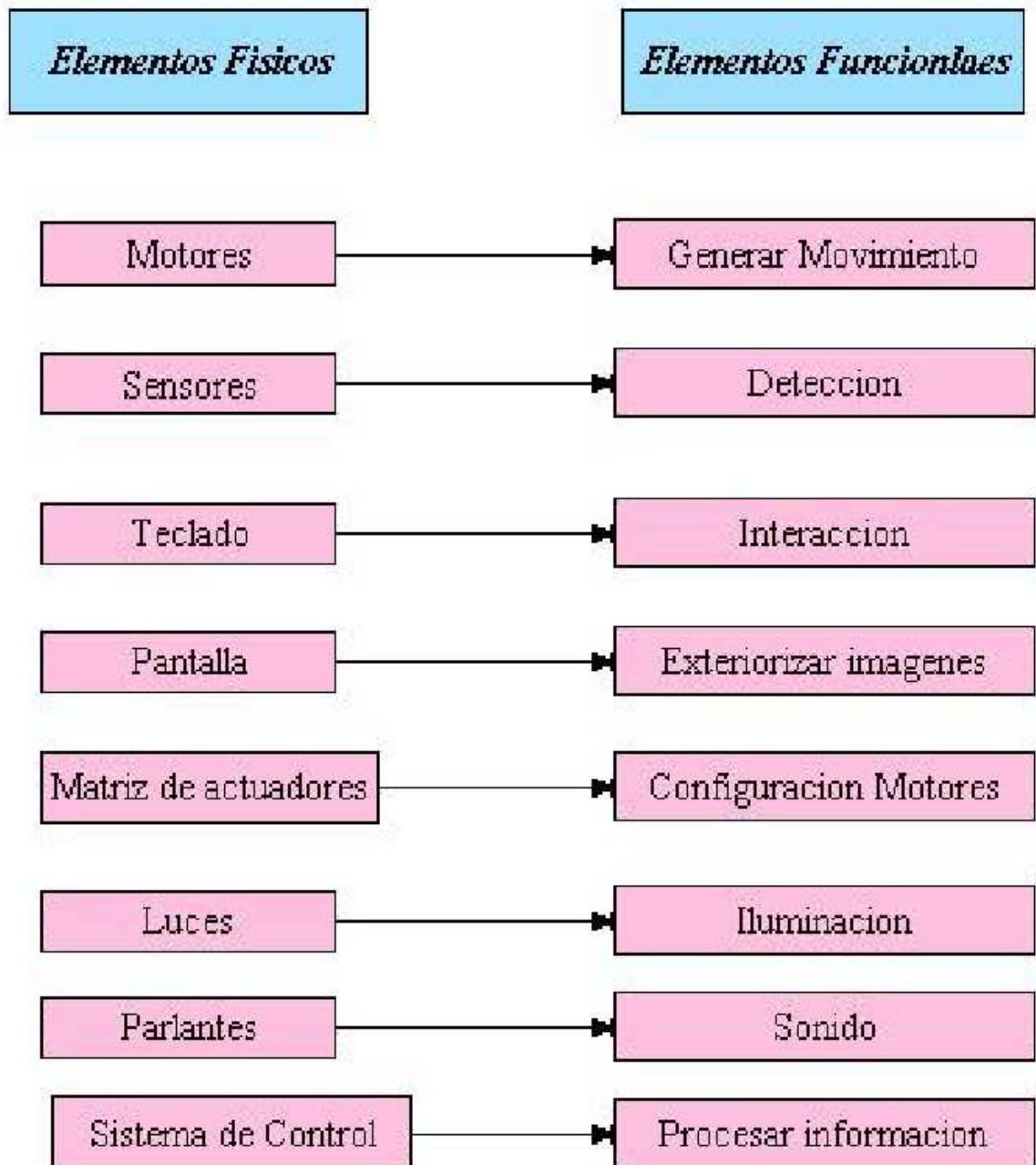
2.1. ESQUEMA DE PRODUCTO

Figura1. Esquema de producto



2.2. ELEMENTOS FÍSICOS Y FUNCIONALES

Figura 2. Elementos físicos y funcionales



2.3. DISTRIBUCIÓN GEOMÉTRICA DE LOS COMPONENTES

Figura 3. Distribución geométrica

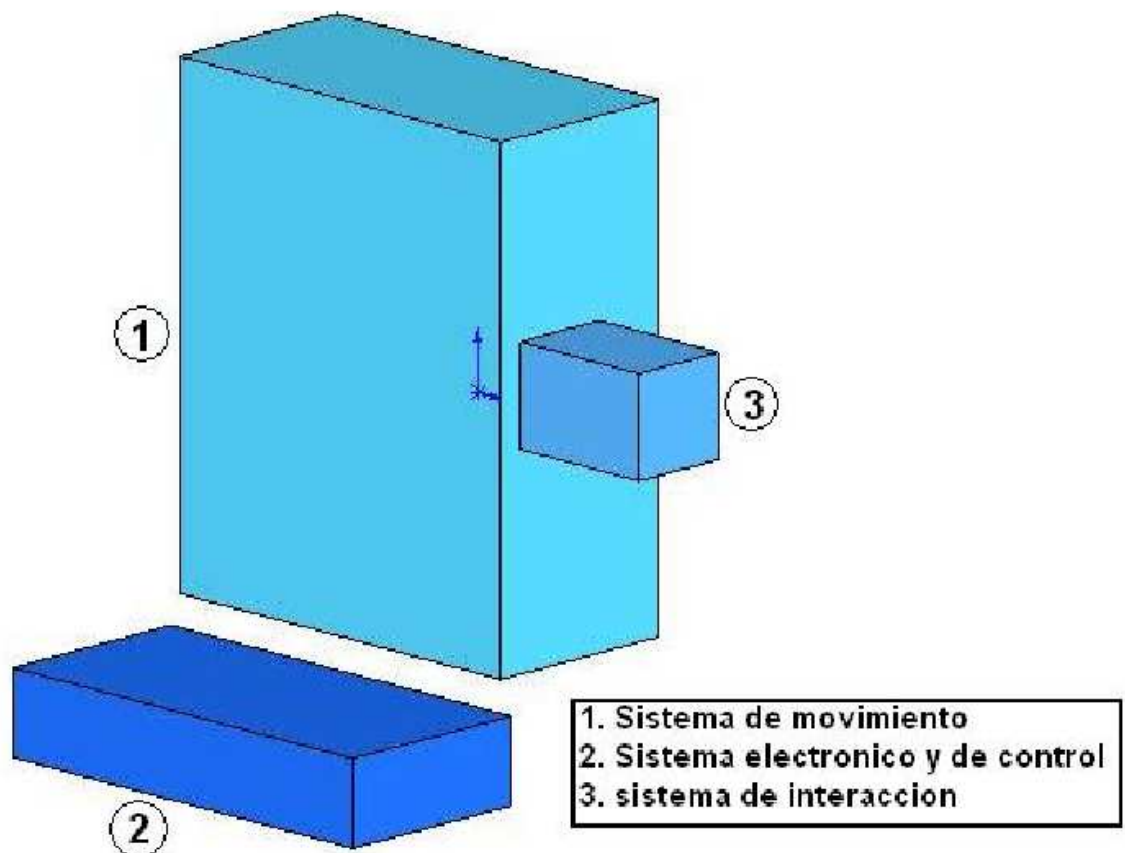
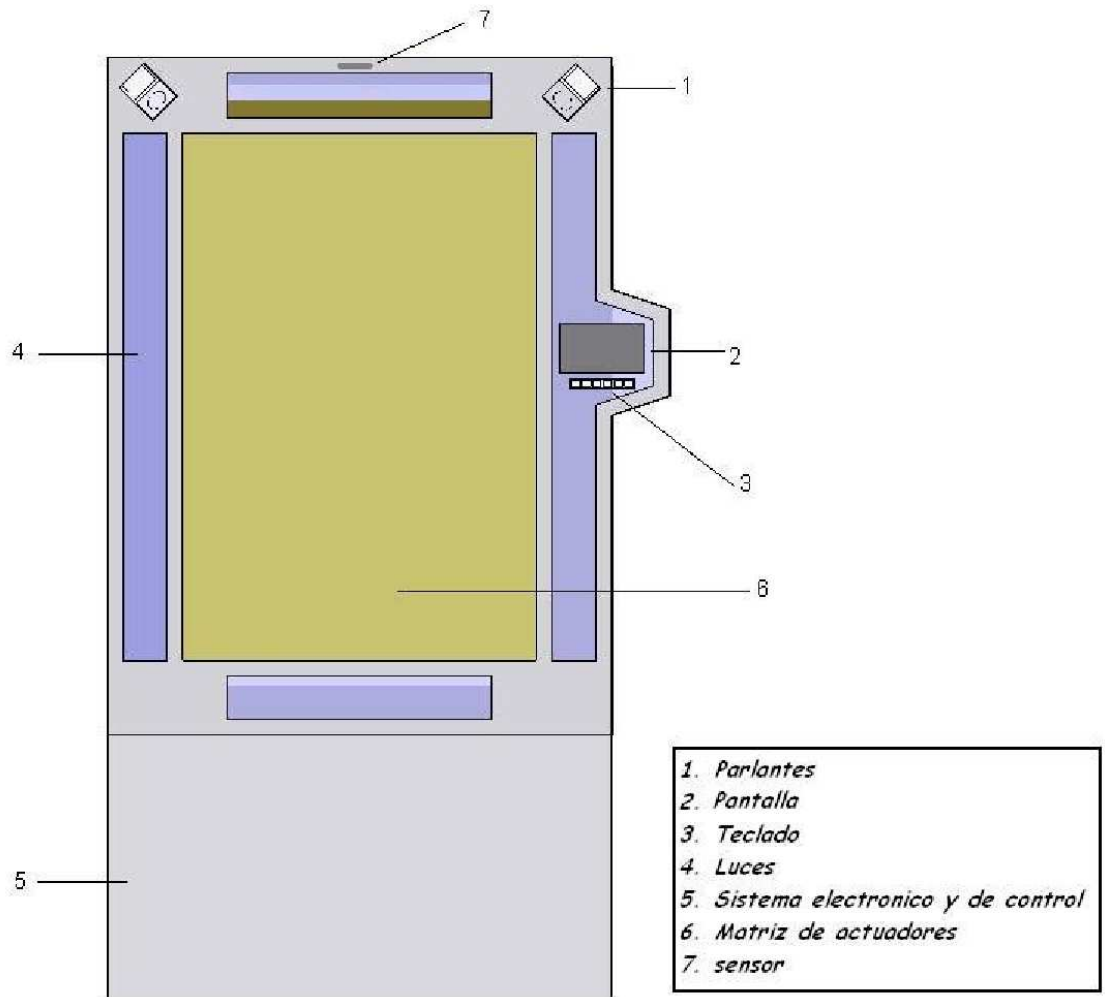


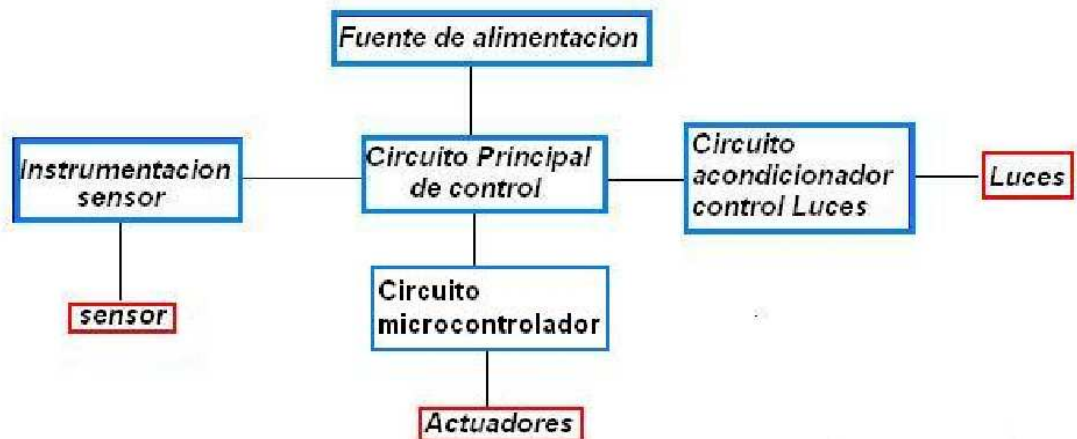
Figura 4. Distribución de componentes



2.4. ARQUITECTURA DEL SISTEMA ELECTRÓNICO

- Instrumentación para el sensor con el fin de acondicionar la señal proporcionada por éste.
- Fuente reguladora de voltaje de alimentación.
- Circuito de acondicionamiento para el control de efectos de iluminación
- Circuito de amplificación de audio.
- Circuito driver con PIC para los servomotores.
- Circuito de control principal que es el encargado de recibir, procesar y enviar las señales adecuadas a los actuadores.

Figura 5. Arquitectura del sistema electrónico



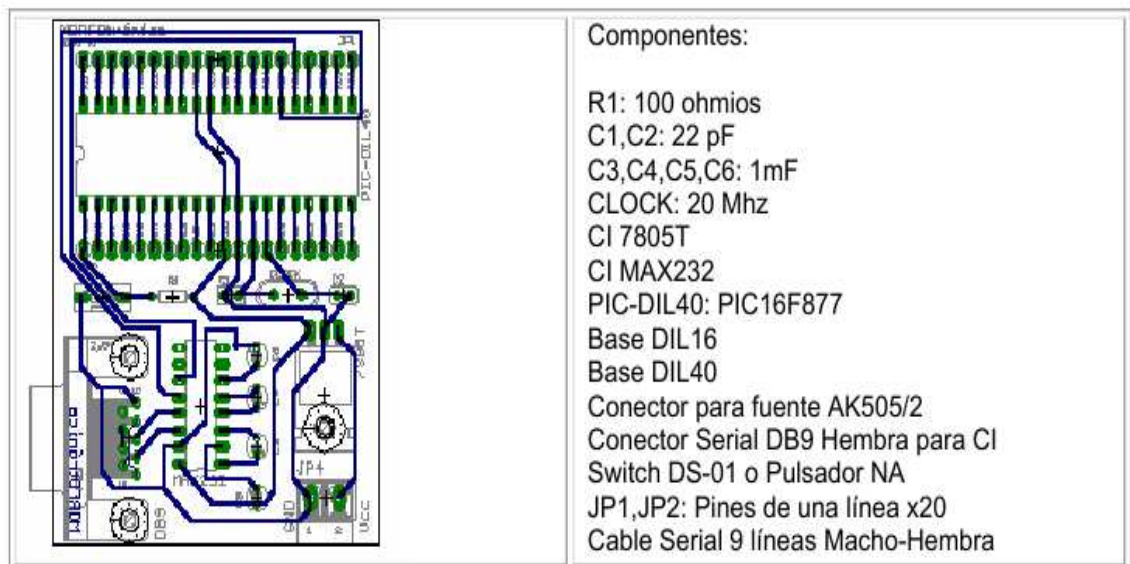
2.4.1. Circuito principal de control. Durante el proceso de selección de conceptos el grupo de diseño llegó a la conclusión que el sistema electrónico y de control, sería un sistema embebido para el almacenamiento y procesamiento de la información.

Este sistema de desarrollo esta bajo la plataforma PC-Linux y será el encargado del control total del sistema y abarcara las funciones de reproducción de audio, iluminación, control de interacción, vídeo, control de actuadores y demás funciones inherentes a la valla.

Esta plataforma esta controlada por un sistema operativo SUSE 10.0, el sistema desarrollado para el control de todos los componentes es programado mediante el sistema de desarrollo KDevelop 3.0 y librerías QT 3.0 haciendo uso del lenguaje C++ una programación orientada a objetos. Este sistema es el encargado de controlar la interacción con el usuario mediante el touch screen.

2.4.2. Circuito microcontrolador. El microcontrolador PIC16F877 de Microchip, cumple con las especificaciones establecidas anteriormente al realizar el Análisis de arquitectura del sistema electrónico, este cumple con la función de driver de los servomotores y a su vez está en constante comunicación por medio del puerto serial con la Tarjeta de control principal.

Figura 6. Diagrama esquemático electrónico

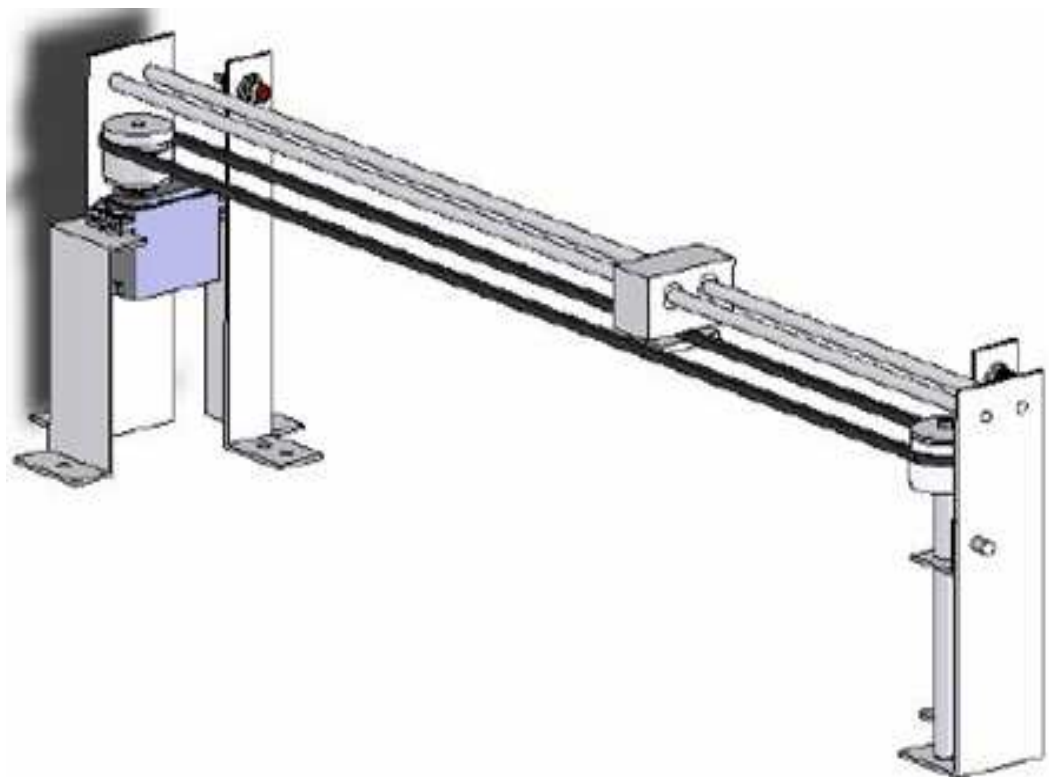


2.5. SISTEMA DE MOVIMIENTOS

La valla robótica posee diferentes subsistemas de movimientos los cuales, ayudado por un grupo de creativos, permiten configurar los diferentes movimientos requeridos para la publicidad. Los movimientos son los siguientes:

2.5.1. Movimiento Lineal

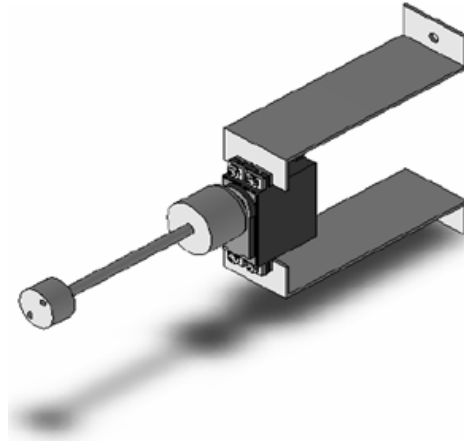
Figura 7. Movimiento lineal



Para este sistema se diseñó un mecanismo que permite el desplazamiento en el eje X y Y dentro de los límites establecidos por la pauta, configurándose fácilmente para cualquier distancia. Este mecanismo consta de un servomotor modificado sostenido por una base ajustable. Este, a través de un sistema banda-polea, genera el movimiento de la figura que a su vez se ajusta a la banda por un acople mecánico. Este movimiento es delimitado por dos sensores finales de carrera para poder determinar su distancia y ubicación.

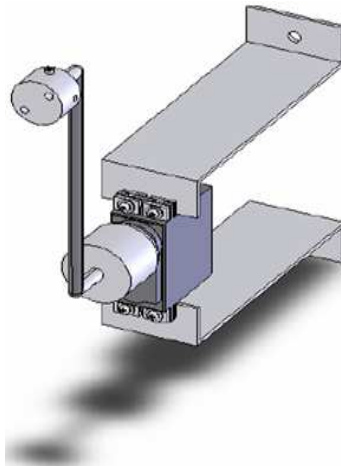
2.5.2. Movimiento rotacional. En este sistema existen dos tipos de movimientos:

Figura 8. Movimiento rotacional concéntrico



- Movimiento rotacional concéntrico como su nombre lo indica la figura describe un movimiento Sobre el mismo eje del servomotor. Este mecanismo consta de un servomotor modificado sostenido por una base ajustable, la figura esta sujeta directamente al eje del actuador por medio de un acople mecánico.

Figura 9. Movimiento Rotacional Excéntrico

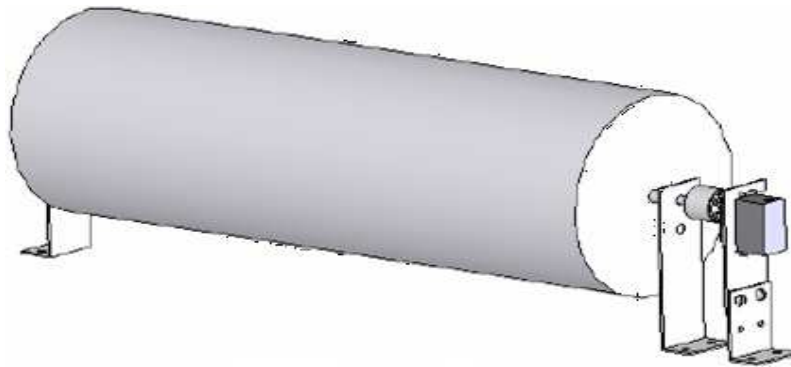


- Movimiento rotacional excéntrico, por el contrario al concéntrico, la figura describe un movimiento por fuera del eje del servomotor. Este mecanismo consta de un servomotor modificado sostenido por una base ajustable, la figura esta

sujeta a un eje el cual le da la excentricidad al movimiento, este eje se sujeta al actuador por medio de un acople mecánico.

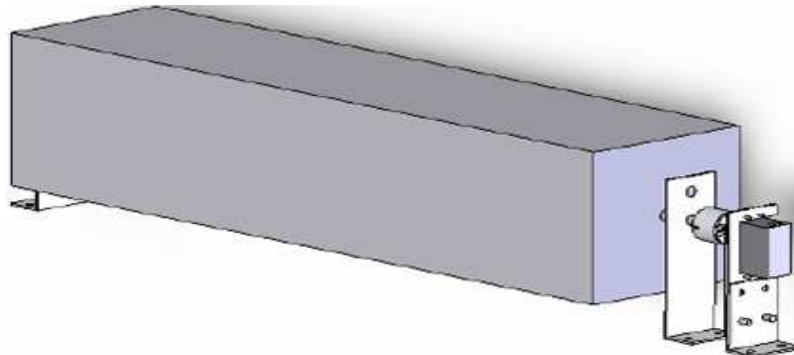
2.5.3. Movimientos especiales.

Figura 10. Cilindro



- Cilindro: está conformado por un servomotor modificado sostenido por una base ajustable, el actuador se acopla a un cilindro por medio de acoples mecánicos donde este a su vez está soportado por un eje que descansa en dos bases ajustables, las cuales permiten que el cilindro gire constantemente.

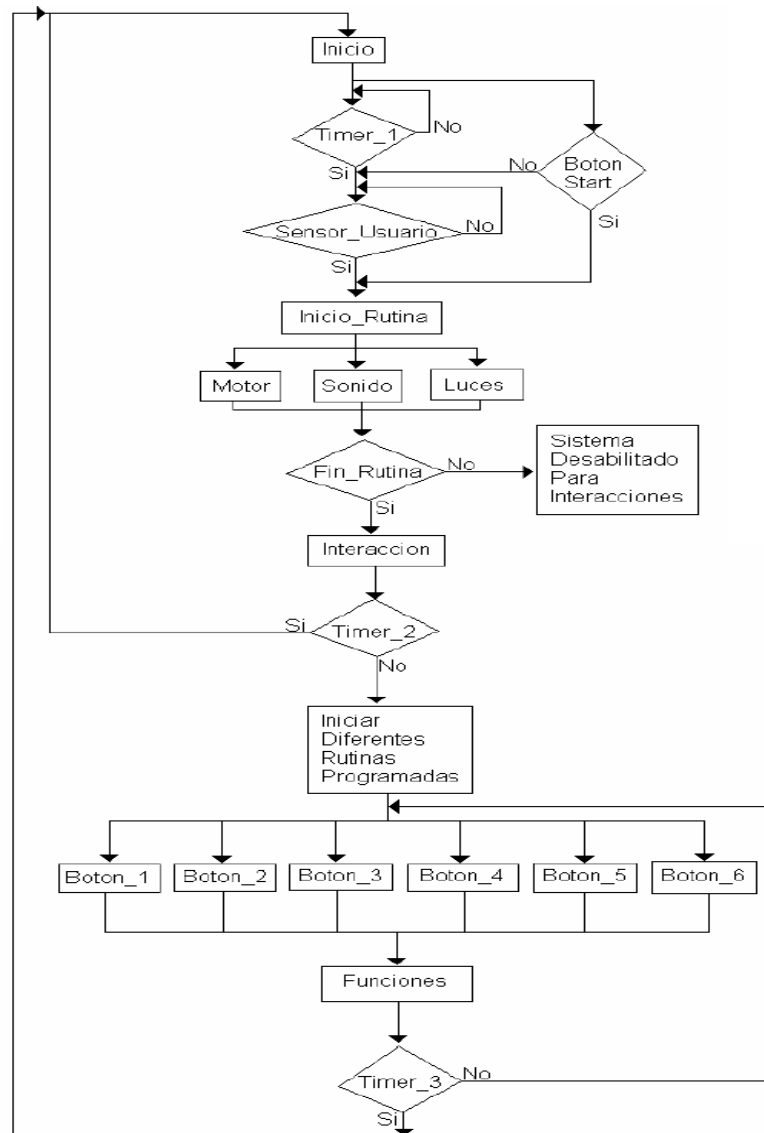
Figura 11. Cubo



- El cubo posee el mismo sistema que el cilindro la diferencia radica en que mientras que el cilindro gira constantemente, el cubo hace pausas en diferentes espacios de tiempo de acuerdo como lo requiera el cliente, esto se logra por medio del software de programación.

2.6. DIAGRAMA DE FLUJO

Figura 12. Diagrama de flujo



En la Figura 12, se muestra en el diagrama de flujo como se implementan las rutinas en el software para una rutina de la valla.

Al iniciar, el sistema espera a que el sensor de proximidad detecte al usuario, o que éste presione el botón de inicio; una vez ocurrido esto se inicia la rutina con sonido, luces y movimientos, el sistema espera hasta que finalice la rutina (no antes) para que después el usuario pueda interactuar con la valla. El usuario

puede presionar cualquier botón el cual tendrá una rutina diferente de acuerdo a su programación previa (definida por el cliente), el sistema esperara un determinado tiempo desde la última interacción con el usuario, se reiniciará y estará lista para otra rutina.

2.7. ESPECIFICACIONES FINALES

La figura 13 muestra la apariencia final de la valla ANIMAK.

Figura13. Animak



Tabla 1. Especificaciones finales

#	Métrica	Unidades	Valores Finales
1	Altura	cm	210
2	Largo	cm	110
3	Ancho	cm	65
4	Peso	Kg.	150
5	Facilidad de instalación y mantenimiento	List1	Excelente
6	Ergonomía	List2	Excelente
7	Estética	List3	Excelente
8	Interacciones	Nº botones de interacción	6
9	Capas de material publicitario	Nº capas	3
10	Alto de la publicidad	cm	120
11	Largo de la publicidad	cm	80
12	Movimientos de traslación	GDL-Traslación	3
13	Máxima distancia de traslación	cm	120
14	Máxima velocidad de traslación	cm/s	2
15	Movimientos de rotación	GDL-rotación	2
16	Máximo ángulo de rotación	grados	350
17	Máxima velocidad de rotación	grados/s	60

Figura 14. Esquema funcional

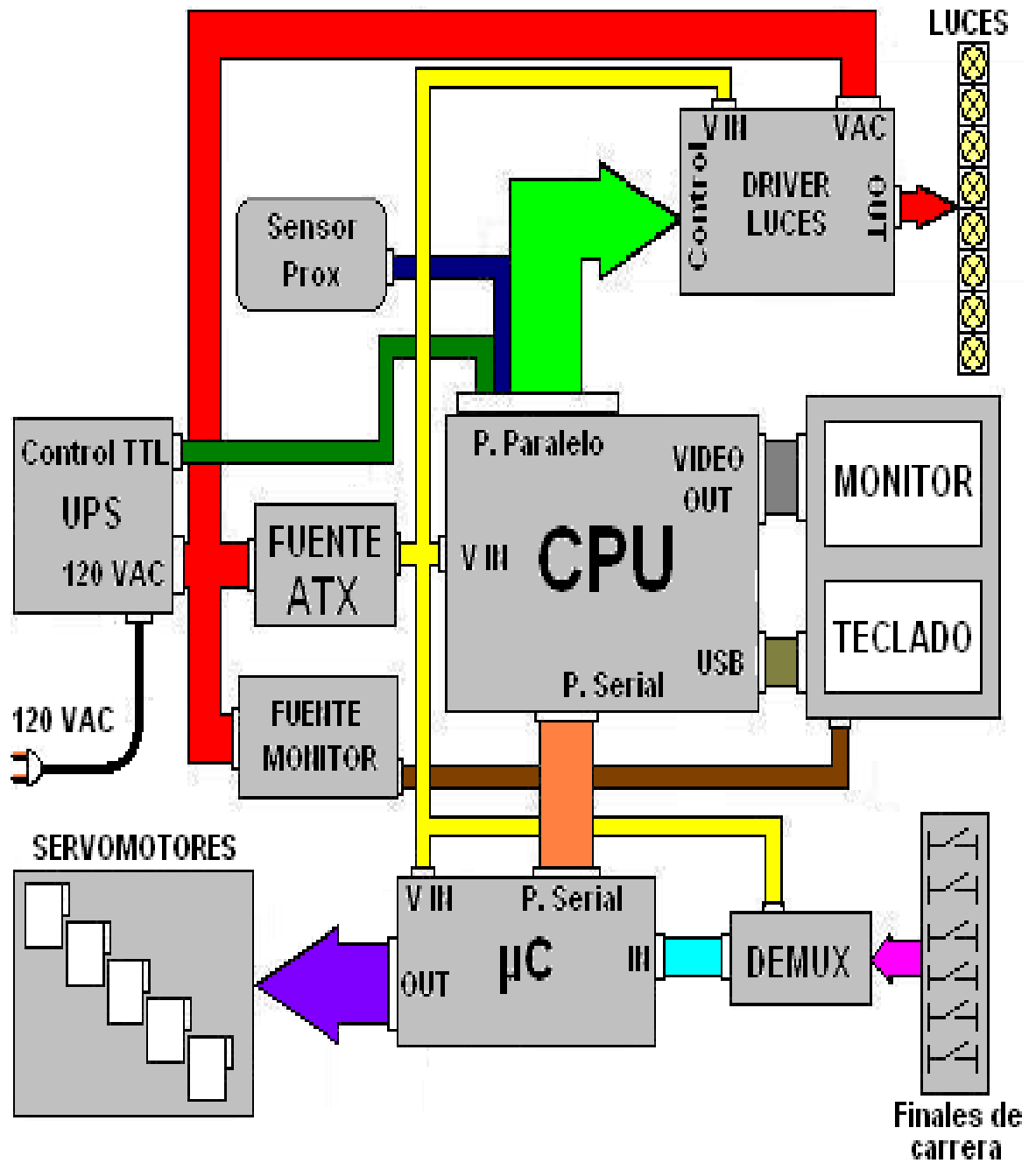









Tabla 2. Tabla de convenciones del esquema funcional

COLOR	NUMERO DE LINEAS	ORIGEN	DESTINO	TIPO DE CABLE / CONECTOR
	2	UPS	FUENTES DE PODER Y DRIVER DE LUCES	DUPLEX #14 / CONECTOR PARA 120 VAC
	2	TARJETA DE VIDEO CPU	TOUCH SCREEN	CABLE BLINDADO PARA SEÑAL DE VIDEO / CONECTOR RCA
COLOR	NUMERO DE LINEAS	ORIGEN	DESTINO	TIPO DE CABLE / CONECTOR
	MINIMO 3 MAXIMO 60	TARJETA DE CONTROL (PIC)	SERVOMOTORES	CABLE SENCILLO DE COBRE # 16
	4	FUENTE ATX (5-12Vcc)	TARJETA DE CONTROL (PIC)	CABLE SENCILLO DE COBRE # 16/ CONECTOR ATX
	21	CPU	CONTROL – DRIVER DE LUCES	CABLE PARA PUERTO PARALELO/DB 25
	2	CPU	CONTROL TTL DE LA UPS	CABLE PARA PUERTO PARALELO/DB 25
	2	CPU	SENSOR DE APROXIMACIÓN	CABLE PARA PUERTO PARALELO/DB 25

Continuación Tabla 2

COLOR	NUMERO DE LINEAS	ORIGEN	DESTINO	TIPO DE CABLE / CONECTOR
	2	FUENTE DEL TOUCH SCREEN	TOUCH SCREEN	DUPLEX # 16 / CONECTOR PARA 12 Vcc
	9	CPU	TARJETA DE CONTROL (PIC)	CABLE PARA PUERTO SERIAL / DB 9
	8	SENSORES FINALES DE CARRERA	DEMUX	CABLE DE RED / CONECTOR DE RED
	6	DEMUX	TARJETA DE CONTROL (PIC)	CABLE SENCILLO DE COBRE # 16
	4	CPU	TOUCH SCREEN	CABLE USB / CONECTOR USB

3. ESPECIFICACIONES DE SOFTWARE

De acuerdo con los datos obtenidos en el análisis de requisitos, se determinaron las características funcionales del software. A continuación se describe de forma detallada el funcionamiento del software y la manera como se comunicara con el sistema controlador PIC.

3.1. SISTEMA OPERATIVO

Para tener un control mas profundo sobre el comportamiento de todo el sistema, se optó por un sistema operativo de código abierto y con filosofía de software libre como lo es Linux, lo cual nos permite tener una plataforma totalmente disponible para el desarrollo de la aplicación de control.

SuSE 10,0 es la distribución escogida como plataforma principal de desarrollo, debido a su facilidad de instalación y configuración, esta versión cuenta con el software necesario tanto para el desarrollo como para implementar la plataforma de las vallas robóticas. Adicionalmente, esta distribución esta respaldada por Novell Inc. y cuenta con una de las mas amplias comunidades en la red, lo cual brinda una actualización constante y un excelente soporte técnico.

Este sistema operativo (SO), proveerá las herramientas para realizar la comunicación a nivel de maquina, con los puertos de comunicación y la interfaz. Manejará las interrupciones generadas por la interfaz gráfica y la interacción del usuario, comunicará todos los dispositivos del PC con el sistema de control y proporcionará los métodos de acceso a los diferentes sistemas de archivos soportados por el dispositivo, con el fin de aprovechar la amplia compatibilidad del SO y dotar a la plataforma de una gran versatilidad.

Proporciona también, todo el entorno de ventanas y el control sobre estas. El entorno de ventanas escogido para el desarrollo fue el KDE, el cual es el que viene por defecto en la distribución. Este entorno se basa en las librerías QT, las cuales dotan la plataforma con una gran calidad visual.

3.2. FUNCIONAMIENTO GENERAL DEL SOFTWARE

La denominación para el software en fase de desarrollo fue Frankie. Este deberá ser capaz de controlar las comunicaciones con los diferentes sistemas, administrar los recursos de medio del PC (Sistema de archivos, pantalla, sonido, etc.) y controlar el flujo de reproducción e interacción, con el fin que no existan procesos corriendo de manera descontrolada. Para ello, cuenta con 4 relojes internos que determinan:

- 1. El tiempo entre los frames animados.
- 2. El intervalo para la verificación de sensores de interacción, esto permite que la valla determine cuanto tiempo ha pasado desde su ultima interacción con el usuario, para así reiniciar la secuencia de frames. Esto le da a la valla, la capacidad de enviar secuencias específicas de animación y así continuar con la siguiente interacción.
- 3. El intervalo para el escaneo de proximidad, este determina cada cuanto tiempo se verifica el estado del sensor con el fin de que no sea excesivo en entornos de alto flujo de personas.
- 4. El tiempo de inicio de sensores.

El tiempo de reproducción de las pautas audibles, será controlado directamente por el tiempo de la animación.

El sistema puede desplegar una interfaz adicional de encuestas al final de cada animación, estas respuestas pueden ser extraídas del sistema mediante memorias USB, las cuales solo serán detectadas por el sistema en el modo de administración. Todo los relojes son configurables mediante esta interfaz de administración con el fin de adaptar cada valla para un entorno específico.

El sistema se inicia automáticamente a primera hora, esto se logra gracias a los estándares actuales de PC, en el cual la norma ACPI, cuenta directamente desde la misma BIOS del equipo la configuración para el inicio programado.

La animación es generada por los ingenieros de Kafka, en formato xml, este estándar permite la correcta lectura de la información por parte de Frankie.

A continuación se muestra un ejemplo de animación:
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

Pista 0
0
true
ninguna
0
O1E42
A1E13
A2E13
A3E13
R1E80
S1E30
P1E13
C1E80
C2E80
B1E0
B2E0
B3E0
B4E0
B5E0
B6E0
B7E0
.
.
.
.
.
.
.
.
99
O1E34
A3E23
A2E21
R1E80
S1E30
C1E80
C2E80
B1E0
B2E0
B3E0
B4E0

B5E0
B6E0
B7E0

Las cadenas de acción están compuestas por el tipo de actuador, el número y el estado en él debe estar en el frame. Estos datos son tomados por Frankie e interpretados en direcciones y posiciones de los servomotores y enviados al sistema PIC o al puerto de control de luces, según sea el caso.

Implementación del código de interpretación y envío de datos:

```
if(temp=="B"){  
  if(temp=="B1")dato=1;  
  if(temp=="B2")dato=2;  
  if(temp=="B3")dato=4;  
  if(temp=="B4")dato=8;  
  if(temp=="B5")dato=16;  
  if(temp=="B6")dato=32;  
  if(temp=="B7")dato=64;  
  if(temp=="B8")dato=128;  
  .  
  .  
  .  
  .  
  .  
  Paralelo->enviarByte(dato);  
  else{  
    .  
    .  
    .  
    Accion=true;  
    if(temp=="A1"){dato=0x25;Serial->enviarByte(dato);return;}  
    if(temp=="A2"){dato=0x26;Serial->enviarByte(dato);return;}  
    if(temp=="A3"){dato=0x27;Serial->enviarByte(dato);return;}  
    if(temp=="A4"){dato=0x28;Serial->enviarByte(dato);return;}  
    if(temp=="P1"){dato=0x29;Serial->enviarByte(dato);return;}  
    if(temp=="P2"){dato=0x2A;Serial->enviarByte(dato);return;}  
    if(temp=="C1"){dato=0x2B;Serial->enviarByte(dato);return;}  
    if(temp=="C2"){dato=0x2C;Serial->enviarByte(dato);return;}  
    if(temp=="R1"){dato=0x2D;Serial->enviarByte(dato);return;}  
    if(temp=="R2"){dato=0x2E;Serial->enviarByte(dato);return;}  
    if(temp=="R3"){dato=0x2F;Serial->enviarByte(dato);return;}
```



```

if(temp=="R4"){dato=0x30;Serial->enviarByte(dato);return;}
if(temp=="S1"){dato=0x31;Serial->enviarByte(dato);return;}
if(temp=="S2"){dato=0x32;Serial->enviarByte(dato);return;}
if(temp=="S3"){dato=0x33;Serial->enviarByte(dato);return;}
if(temp=="S4"){dato=0x34;Serial->enviarByte(dato);return;}
if(temp=="O1"){dato=0x35;Serial->enviarByte(dato);return;}
if(temp=="O2"){dato=0x36;Serial->enviarByte(dato);return;}
if(temp=="O3"){dato=0x37;Serial->enviarByte(dato);return;}
if(temp=="O4"){dato=0x38;Serial->enviarByte(dato);return;}
else return;

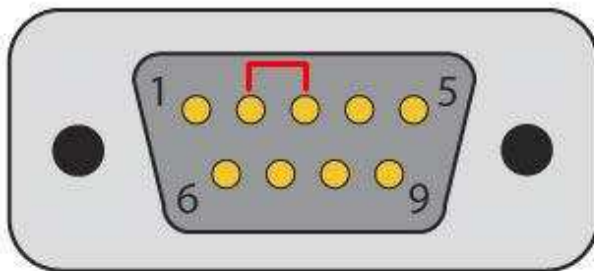
```

3.3. SISTEMAS DE COMUNICACIÓN

3.3.1. Puerto serial

Figura 15. Esquema puerto serial

DB9M Connector



RS232 Pin Out

Pin #	Signal
1	DCD
2	RX
3	TX
4	DTR
5	GND
6	DSR
7	RTS
8	CTS
9	RI

3.3.1.1. Descripción general. Este puerto es una interfaz de comunicaciones entre ordenadores y periféricos el cual envía y recibe información BIT por BIT. Un puerto serial posee un conector estándar y trabaja con un protocolo que permite la conexión de dispositivos al computador. Se denomina “serial” porque el puerto serie “serializa” los datos. Esto quiere decir que toma un byte de datos y transmite los 8 bits del byte de uno en uno.

Tabla 3. Distribución de pines en el puerto serial

Nombre	Sentido	Conector 25 pines	Conector 9 pines
TXD	salida	2	3
RXD	entrada	3	2
RTS	salida	4	7
CTS	entrada	5	8
DTR	salida	20	4
DSR	entrada	6	6
DCD	entrada	8	1
RI	entrada	22	9
GND	masa	7	5

Los puertos seriales utilizan un chip especial denominado UART (Universal Asynchronous Receiver/Transmitter). Este chip toma la salida paralela del bus del computador y lo convierte en forma serial, lo que permite la transmisión de los datos a través del puerto. El conector externo para un puerto serial puede ser de 9 o de 25 pines.

En la Tabla 3 se puede observar la distribución de pines en un puerto serial. Las señales TXD y RXD son respectivamente la salida y la entrada de datos serie.

Estos datos se envían de la siguiente manera, por defecto, TXD se encuentra a -12 voltios (nivel alto), cuando se quiere enviar un byte, la línea se pone a 12 voltios (nivel bajo) durante el tiempo de un bit; es el bit de inicio, que marca el comienzo de una transmisión, después se envían los bits del dato, empezando por el menos significativo. El cero (nivel bajo) se representa con 12 voltios, y el uno (nivel alto)

se representa con -12 voltios. Por ultimo se envían uno o dos bits de parada a -12 voltios (nivel alto) antes de iniciar la transmisión del siguiente.

Vemos que los niveles de tensión se encuentran invertidos con respecto a lo que se podría esperar (el 0 son 12 voltios y el 1 son -12 voltios). Esto solo ocurre en estas dos líneas. En el resto, el cero son -12 voltios y el uno son 12.

Todas las líneas están limitadas en corriente a unos 200mA. Esto protege al puerto en caso de que ocurra cualquier cortocircuito. Esta característica también es aprovechada por muchos circuitos simples que se conectan al puerto serie, de modo que se consigue una notable economía de componentes.

Debido a que las líneas TXD y RXD son controladas por el propio puerto serie y no por el procesador, en principio no resulta posible indicar directamente un valor de tensión en TXD o leer el que se encuentre en RXD. Todo lo que podemos hacer es enviar un byte al puerto para que este lo envíe por si mismo en el formato indicado. Sin embargo, en la practica, las UARTs disponen de un bit que, al activarlo, envían un cero (+12 voltios) en dicho pin.

3.3.1.2. Especificación de uso. Este puerto permite la comunicación con el microcontrolador, el cual es el encargado de administrar los servomotores.

Los paquetes de datos enviados deben contener la dirección del dispositivo y el valor al cual debe ser movido. Para los de rotación continua, estos valores se reducen a 3, parada, giro izquierda y giro derecha; mientras que para los otros movimientos, los servos pueden tener un valor entre 0 y 310. Este puerto también es utilizado para pasar del modo administrador al modo autónomo o viceversa, por lo tanto, es importante contar con una interfaz que constantemente revise la información en el puerto y se lo comunique al programa de control. Para esto, la interrupción generada por la interfaz, debe ser tomada por el software, el cual determina en cual de los 2 estados se encuentra la valla.

Debido a que la velocidad de transmisión de los datos es muy importante para poder obtener una animación fluida, la comunicación serial será realizada asincrónicamente, es por esto que el sistema debe informar al controlador, el momento en el que los datos empiezan a ser enviados a él. De ahí en adelante, los datos deben ser enviados en los paquetes preestablecidos para no enviar información incorrecta.

En Linux, el entorno determinado para el desarrollo del producto, representa el hardware dentro del sistema como un archivo. Mediante el estándar POSIX, es posible acceder a la información, así como enviarla en este puerto, como si se tratara de un archivo común. Este estándar provee al sistema, de unas herramientas totalmente eficientes, seguras y comprobadas, para el uso de la comunicación serial.

El archivo que describe los puerto seriales en un sistema openSUSE/Linux se denomina `ttyS*`, donde '*' es el numero del puerto. Así, haciendo una analogía con un sistema Windows, el `ttyS0` se refiere al COM1 y así sucesivamente. Estos archivos se encuentran en el directorio `/dev/` del sistema raíz de directorios. Para acceder a ellos es necesario tener privilegios de root o superusuario, y es fundamental la adecuada configuración del puerto. Para acceder a un puerto serial, simplemente se debe abrir el archivo de dispositivo correspondiente mediante la función `open()`.

Tabla 4. Estructura Termios

Miembros:	Descripción:
<code>c_cflag</code>	Opciones de control
<code>c_lflag</code>	Opciones de línea
<code>c_iflag</code>	Opciones de entrada
<code>c_oflag</code>	Opciones de salida
<code>c_cc</code>	Caracteres de contro
<code>c_ispeed</code>	Velocidad de entrada
<code>c_ospeed</code>	Velocidad de salida

Las dos funciones POSIX más importantes son `tcgetattr()` y `tcsetattr()`, las cuales obtienen y establecen los atributos de terminal; se debe proporcionar un apuntador a la estructura termios que contiene todas las opciones de comunicaciones y control.

`c_cflag` contiene dos opciones que siempre se deben habilitar: `CLOCAL` y `CREAD`. Esto asegura que el programa no se “adueñe” del puerto, y que el controlador de interfaces lea los datos que llegan. Las constantes de baud rate (`B9600`, `B38400`, etc.) se usan para interfaces viejas que carecen de los miembros `c_ispeed` y `c_ospeed`. Las opciones deben asignarse mediante punteros y operadores lógicos de bits (`NOT`, `OR`, etc..)

Las funciones `cfsetospeed()` y `cfsetispeed()` establecen el baud rate, Mientras que la longitud del dato debe determinarse mediante enmascaramiento, ya que no existe una funcion explicita para ello.

La paridad se establece manualmente mediante:

- Sin paridad: `options.c_cflag &= ~PARENB;`
- Paridad par: `options.c_cflag |= PARENB; options.c_cflag &= ~PARODD;`
- Paridad impar: `options.c_cflag |= PARENB; options.c_cflag |= PARODD;`

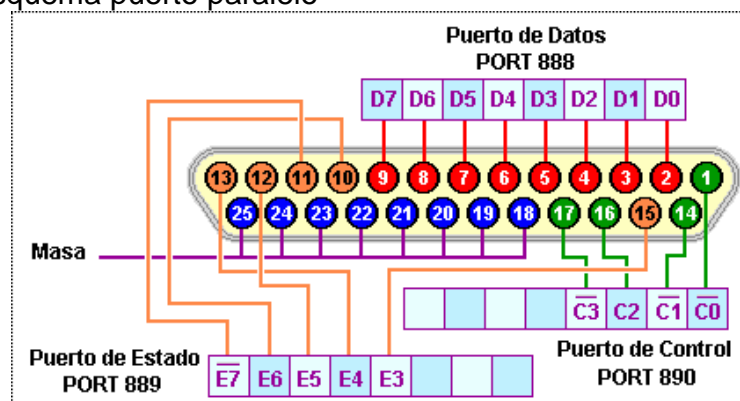
Tabla 5. Opciones de control

Constante:	Descripción:
B50	50 <u>baud</u>
B300	300 <u>baud</u>
B600	600 <u>baud</u>
B1200	1200 <u>baud</u>
B4800	4800 <u>baud</u>
B9600	9600 <u>baud</u>
B19200	19200 <u>baud</u>
B38400	38400 <u>baud</u>
B57600	57,600 <u>baud</u>
B115200	115,200 <u>baud</u>
EXTA	<u>External rate clock</u>
EXTB	<u>External rate clock</u>
CSIZE	<u>Bit mask for data bits</u>
CS5	5 data bits
CS6	6 data bits
CS7	7 data bits
CS8	8 data bits
CSTOPB	2 <u>stop bits</u> (1 <u>otherwise</u>)
CREAD	<u>Enable receiver</u>
PARENB	<u>Enable parity bit</u>
PARODD	<u>Use odd parity instead of even</u>
HUPCL	<u>Hangup (drop DTR) on last close</u>
CLOCAL	<u>Local line - do not change</u>
LOBLK	<u>"owner" of port</u>
CNEW_RTSCTS	<u>Block job control output</u>
CRTSCTS	<u>Enable hardware flow control</u>

La función `read()` lee desde un descriptor de archivo: la instrucción “`ssize_t read(int fd, void *buf, size_t count)`”, intenta leer `count` bytes en el buffer que empieza en `buf` del archivo de referencia a `fd`. Cuando el puerto opera en modo crudo, cada llamada a `read()` regresará muchos caracteres que están disponibles en el buffer de entrada serial. Si no hay caracteres disponibles, la llamada se bloquea (esperando) hasta que lleguen, que expire un intervalo de tiempo, u ocurra un error.

3.3.2. Puerto paralelo

Figura 16. Esquema puerto paralelo



3.3.2.1. Descripción general. Un puerto paralelo tiene ocho líneas. El puerto paralelo de las computadoras de acuerdo a la norma Centronic, está compuesto por un bus de comunicación bidireccional de 8 bits de datos, además de un conjunto de líneas de protocolo. Las líneas de comunicación cuentan con un retenedor que mantiene el último valor que les fue escrito hasta que se escribe un nuevo dato; las características eléctricas son:

- Tensión de nivel alto: 3.3 o 5 V.
- Tensión de nivel bajo: 0 V.
- Intensidad de salida máxima: 2.6 mA.
- Intensidad de entrada máxima: 24 mA.

El sistema operativo gestiona las interfaces de puerto paralelo con los nombres LPT1, LPT2 y así sucesivamente, las direcciones base de los dos primeros puertos es:

- LPT1 = 0x378.
- LPT2 = 0x278

La estructura consta de tres registros: de control, de estado y de datos.

- El **registro de control** es un bidireccional de 4 bits, con un bit de configuración que no tiene conexión al exterior, su dirección en el LPT1 es 0x37A. Este registro permite controlar la transferencias de información con la impresora, y puede ser escrito y leído desde el microprocesador.

- El bit 4 (IRQ)
- El bit 3 (DSL)
- El bit 1 (ALF)
- El bit 0 (STR)
- Los bits 5, 6 y 7 no se utilizan.

- El **registro de estado**, se trata de un registro de entrada de información de 5 bits, su dirección en el LPT1 es 0x379. El registro de estado indica la situación actual de la impresora conectada al puerto, de acuerdo con los niveles de tensión que tengan las líneas ACK, BSY, PAP y OF/ON , lo que permite controlar el comportamiento de la impresora. El significado que tienen los bits de este registro es el siguiente:

- El bit 7 (/Busy)
- El bit 6 (ACK)
- El bit 5 (PAP)
- El bit 4 (OF/ON)
- El bit 3 (ERR)

Los bits 0,1 y 2 no se utilizan. Su nomenclatura esta relacionada con su uso especifico para el manejo de impresoras.

- El **registro de datos**, se compone de 8 bits, es bidireccional. Su dirección en el LPT1 es 0x378.

3.3.2.2. Especificación de uso. Este puerto es utilizado como interfaz de control para el sistema de iluminación de la valla. Por cada uno de los 8 pines de datos de este puerto, se puede controlar el estado de encendido y apagado de 1 luz.

Se ha optado por el acceso a este puerto a nivel de maquina, ya que permite tener un mejor control de los datos enviados a este. Para ello, es necesario darle al programa el acceso apropiado al puerto mediante la función `ioperm()`; la cual tramita el acceso ante el sistema y permite la lectura y escritura de una dirección de memoria directamente, con lo cual se eliminan interfaces adicionales.

Para la escritura y lectura en el puerto, las instrucciones `outb()` e `inb()` respectivamente, ofrecen la funcionalidad requerida por el software.

Cada uno de los bits del registro de datos, es utilizado por la valla para controlar una luz, para ello es importante que el puerto sepa qué combinación de luces es la que debe estar activa en cada instante de la reproducción.

3.4. SISTEMA REPRODUCTOR DE AUDIO DIGITAL

3.4.1. Formato MP3. MPEG-1 Audio *Layer 3*, más conocido como MP3 y también por su grafía *emepetrés*, es un [formato de audio digital](#) comprimido con pérdida desarrollado por el Moving Picture Experts Group (MPEG) para formar parte de la versión 1 (y posteriormente ampliado en la versión 2) del formato de vídeo MPEG.

El mp3 estándar es de 44 KHz y un bit-rate de 128 kbps por la relación de calidad/tamaño.

3.4.2. Software de reproducción MPG321. El MPG321 es un reproductor de línea de comando, con toda la calidad de un reproductor de audio digital básico es ideal para sistemas embebidos. Cuenta con un sistema de control remoto diseñado para el desarrollo de interfaces gráficas, su pequeño tamaño y su licencia de libre distribución GNU lo hace perfecto para el desarrollo de la valla.

Su operación es simple; para ejecutar el programa en modo remoto se utiliza la instrucción `mpg321 -r`, con lo cual se abre una línea de comando para la interacción con el programa. Una vez que se está ejecutando con la interfaz de control remoto, hay varios comandos que puede utilizar.

Comandos:(Todos los comandos pueden abreviarse a la primera de carácter únicamente).

LOAD <file> Carga y empieza a reproducir <file>.

JUMP[-] <frames> Adelanta los frames del archivo.

PAUSE Hace una pausa en la reproducción del archivo mp3, si ya se encuentra pausa, se reinicia la reproducción.

STOP Detiene la reproducción de un archivo.

OUT Mpg321 se cierra.

3.5. INTERFAZ GRÁFICA DE USUARIO

El sistema cuenta con 2 esquemas de interfaz, una creada para la interacción con las personas denominada GUI, y otro creado para la administración del sistema GUI_Admin. Las interfaces contarán con un touch screen para el manejo de los eventos del mouse. Es totalmente configurable mediante archivos que especifican desde la cantidad de botones disponibles en la interfaz, así como la distribución de estos en la pantalla. Se utilizan gráficos en formato png, denominados dentro de la estructura de directorios /home/usuario/Datos_Animax/Interfaz/, como b1, b2, etc. Para los botones y para el fondo se utiliza un archivo jpg de 800x600.

La distribución se realiza con un archivo simple de texto que contiene las posiciones XY de cada uno de los botones. Se realizarán teniendo como base las librerías gráficas QT, las mismas utilizadas por el entorno de ventanas KDE.

La interfaz de administración cuenta herramientas de selección y botones, ya que el único medio para el ingreso de datos es el touch screen. Desde esta interfaz se puede cargar un conjunto de botones para la GUI, archivos de audio, encuestas y distribución de elementos de un anunciante nuevo. También se configuran los relojes internos del programa, esto permite que se tenga un control más profundo en el funcionamiento de la valla.

Figura 17. Interfaz de administración



Figura 18. Configuración de relojes

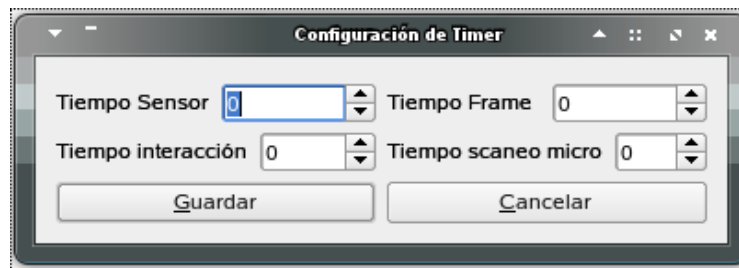


Figura 19. Interfaz gráfica de usuario



4. ESPECIFICACIONES DE HARDWARE

4.1. DESCRIPCIÓN DETALLADA DEL SISTEMA DE CONTROL

Después de depurar los conceptos generados para lograr una comunicación óptima entre el Software controlado por el PC de la valla y el Hardware de la valla, se llegó a la utilización del puerto paralelo y puerto serial del PC para enviar y recibir los datos de control y realimentación que necesita el sistema para su funcionamiento. Por esta razón se requirió del diseño de una etapa de comunicaciones entre el PC y el Hardware de la valla y de ésta manera poder aprovechar al máximo las capacidades de los puertos del PC.

El puerto paralelo del PC se encarga de transmitir los datos de control para la secuencia de 8 luces (o grupos de luces conectadas en paralelo), además de recibir las señales de los sensores finales de carrera para los movimientos lineales, así como también la señal del sensor de movimiento infrarrojo PIR, la señal de la UPS del sistema para generar acción de protección en caso de fallas en el servicio de electricidad, y la señal del Switch de modo que indica cuando el sistema se encuentra en modo de funcionamiento normal o en modo de administrador. Todas las señales son digitales (0 - 5Vcc) y pasan por una rectificación por medio del Driver TTL 74LS244.

Por el puerto serial se envían las señales de control para los 20 servomotores; este envío se realiza por medio de una cadena la cual es interpretada por el sistema de desarrollo del PIC 16F877 de MICROCHIP que también se aloja en la etapa de comunicaciones. Estas señales de control para los servomotores son del tipo PWM y van conectadas directamente a los servomotores mediante conectores diseñados por el fabricante, los cuales llevan además de la señal de control, la alimentación de +5Vcc, y GND independiente para cada servomotor.

4.2. BOARDS O CIRCUITOS IMPRESOS

Para cumplir con los requerimientos exigidos por KAFKA LTDA., el Hardware de la valla ANIMAK debe ser robusto, confiable, seguro, de fácil instalación, reparación y mantenimiento; por este motivo se diseñaron 3 boards o circuitos impresos para lograr integrar las diferentes etapas que tiene el Hardware de la valla.

Después del diseño de estas boards con sus respectivos componentes, se seleccionó la empresa Microcircuitos ubicada en la ciudad de Santiago de Cali, ya que cuenta con la infraestructura necesaria para fabricar las boards para la valla ANIMAK, además de conseguir los componentes de cada board y realizar su montaje.

Para la fabricación de las boards, la empresa Microcircuitos requiere los archivos GERBER, generados en el software de diseño electrónico EAGLE, en los que se diseñó cada una de las boards; y dentro de estos archivos se encuentran todas las especificaciones para lograr obtener físicamente las boards tal y como fueron diseñadas en EAGLE.

4.2.1. Board de comunicaciones (Board 1). Para la interacción de todas las señales y el correcto funcionamiento de cada una de los dispositivos de las etapas del Hardware, se diseñó una tarjeta de comunicaciones de fácil instalación donde se conectan los cables paralelo y serial que van directamente al PC por medio de un conector hembra para puerto paralelo DB 25 y un conector hembra para puerto serial DB 9, además cuenta con 2 conectores hembra de tipo Red para las señales que manejan los 8 drivers de las luces y para las señales de los 8 sensores finales de carrera; para las señales de modo de uso, alimentación de servomotores y señal de UPS se usaron 3 conectores machos tipo regleta-board de 2 pines y para el sensor infrarrojo de aproximación PIR el mismo tipo de conector pero de 4 pines; para las 20 señales de PWM de los servomotores se usó un conector tipo regleta-board macho de 10x2 pines.

Para la alimentación del Hardware, se usó una fuente ATX de 350 Watts, y la recepción de la alimentación se hace por medio de conectores macho de 4 pines para fuente ATX, se usaron 2 de estos para independizar la alimentación del PIC 16F877 y demás dispositivos de control, de los servomotores, ya que estos pueden generar mucho ruido que alteraría el correcto funcionamiento de los sistemas.

Figura 20. Board 1

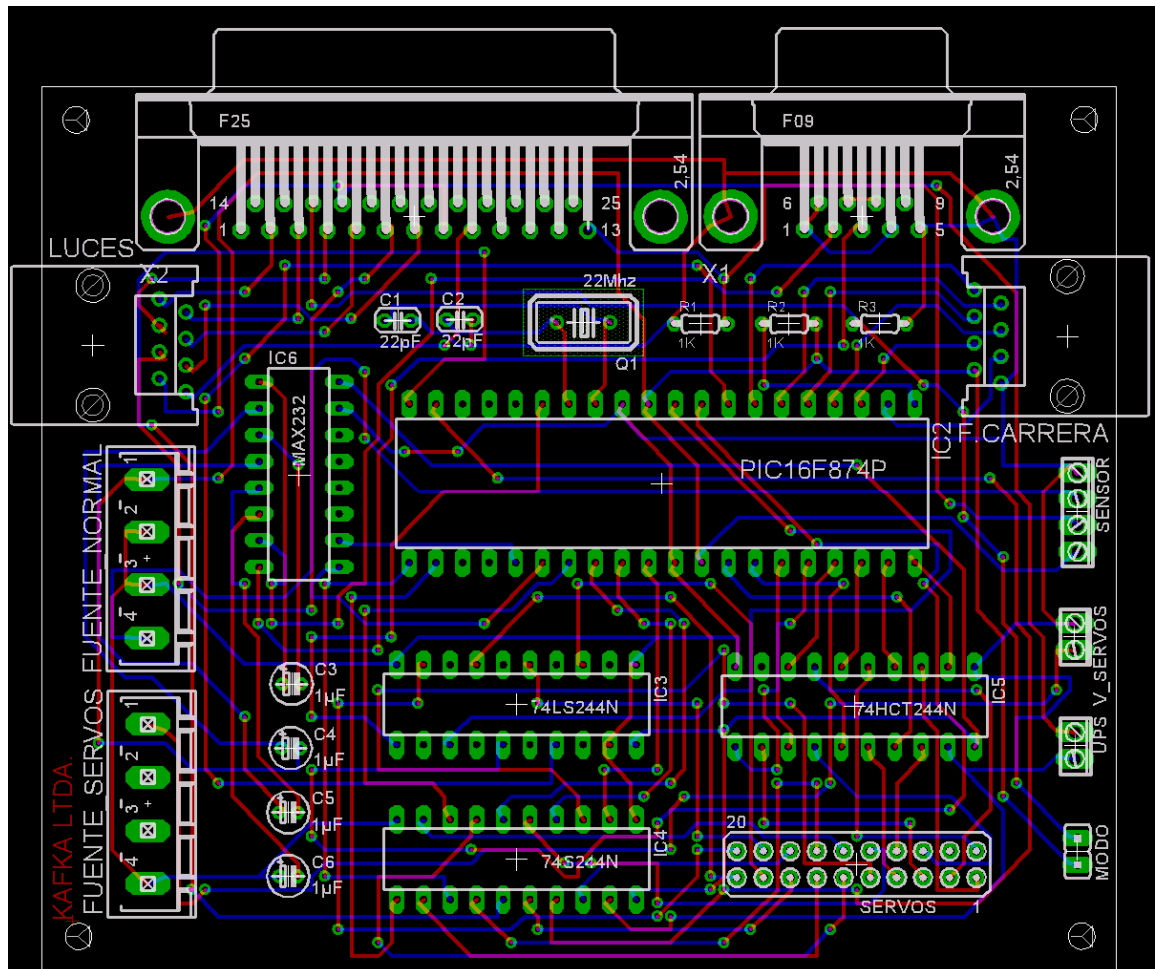


Tabla 6. Tabla de especificaciones de la Board 1

COMPONENTE	REFERENCIA	CANTIDAD	NOMBRE
Microcontrolador	16F877	1	IC2
Driver / Reciver	MAX 232	1	IC6
Cristal	22Mhz	1	Q1
Resistencia	1k Ω , ¼ W	3	R1, R2, R3
Condensador cerámico	22pF	2	C1, C2
Condensador electrolítico	1 μ F	4	C3, C4, C5, C6
Buffer / Line driver	74 LS 244	3	IC3, IC4, IC5
Conector para puerto paralelo hembra	DB 25	1	X2
Conector para puerto serial hembra	DB 9	1	X1
Conector hembra para cable de red	RJ 45	2	LUCES, F.CARRERA
Conector de corriente hembra fuente ATX	AMP 1-480424-0	2	FUENTE SERVOS, FUENTE NORMAL
Conector hembra de seguridad de 2 pines	Conector hembra de tornillo de 2 pines	2	UPS, V_SERVOS
Conector macho de 2 pines tipo regleta	Sección de regleta macho de 2 pines	1	MODO
Conector hembra de seguridad de 4 pines	Conector hembra de tornillo de 4 pines	1	SENSOR
Conector macho de 10 pines tipo regleta	Sección de regleta macho de 10 pines	2	SERVOS

4.2.2. Board de conexiones para servomotores (Board 2). Esta tarjeta es una simple extensión de la tarjeta de comunicaciones y se diseñó para facilitar la instalación de los 20 servomotores y tener una forma organizada para su distribución en la matriz de actuadores. Como los servomotores requieren 3 señales que son +VCC, GND y PWM, entonces se eligió un conector tipo regleta-board de 60 pines macho para conectar directamente los servomotores dependiendo de la programación de la valla, siendo un caso extremo el uso de la capacidad total de la tarjeta (20 servomotores); también cuenta con el mismo tipo de conector instalado en la tarjeta de comunicaciones para la salida de las señales de PWM de 10x2 pines macho.

En esta tarjeta también se alojan los conectores para las 8 señales disponibles para los sensores finales de carrera mediante conectores tipo regleta-board machos de 2 pines cada uno y con su respectivo conector tipo Red para conectarse mediante un cable de Red con la tarjeta de comunicaciones.

Figura 22. Board 2

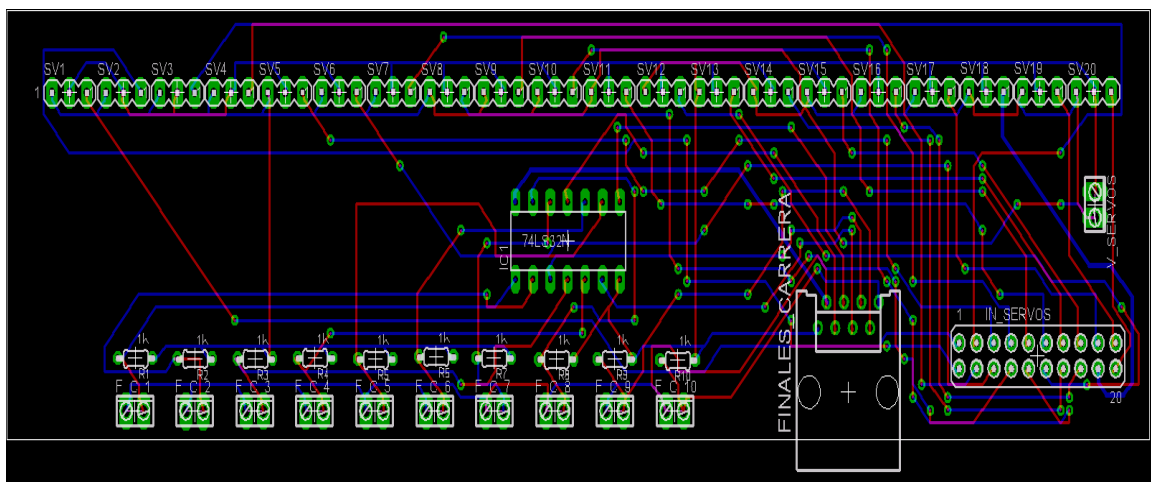


Figura 23. Diagrama esquemático de la Board 2

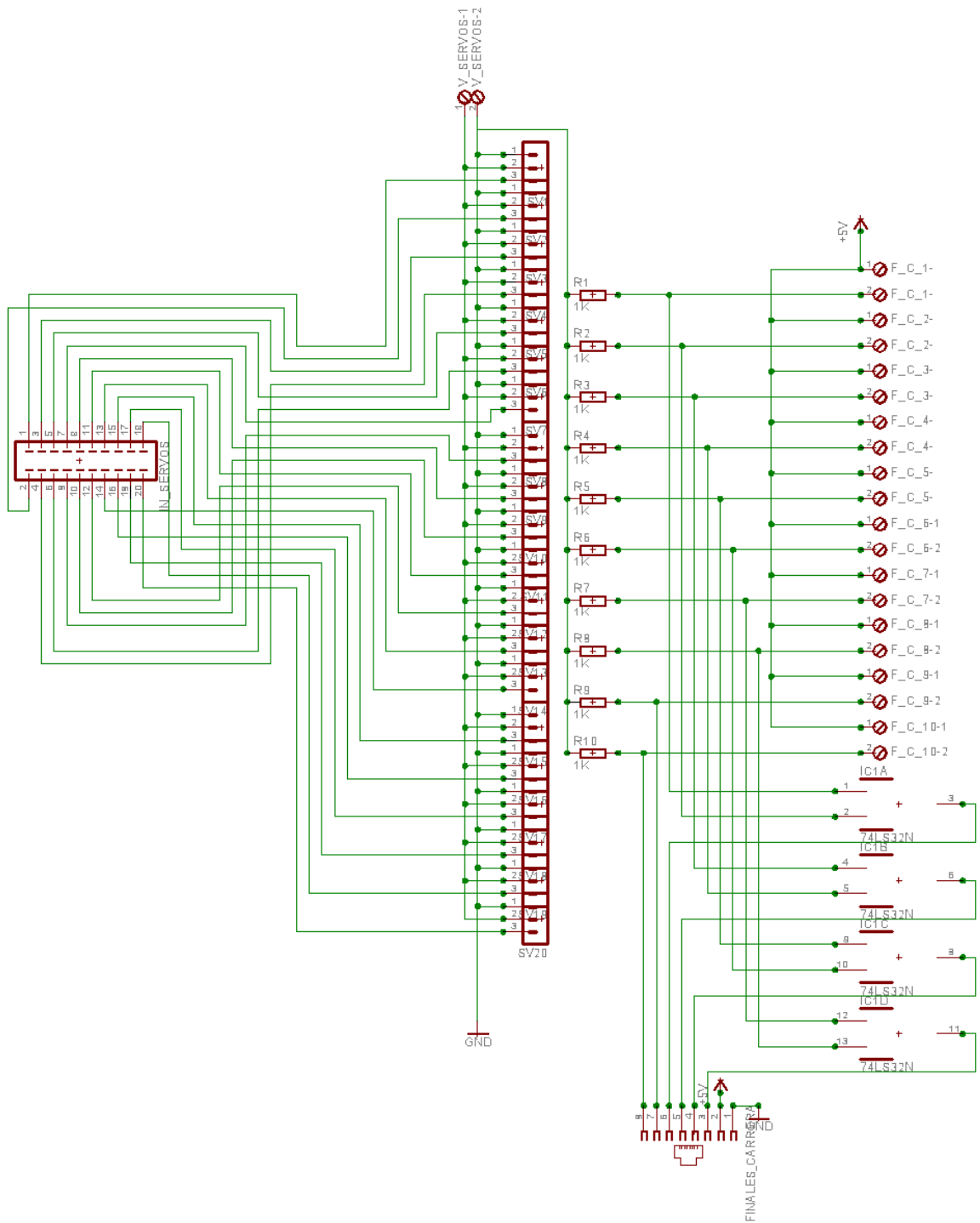


Tabla 7. Tabla de especificaciones de la Board 2

COMPONENTE	REFERENCIA	CANTIDAD	NOMBRE
Circuito integrado TTL OR	74 LS 32 N	1	IC1
Resistencia	1k Ω , ¼ W	10	R1-10
Conector macho de 3 pines tipo regleta	Sección de regleta macho de 3 pines	20	SV1-20
Conector macho de 10 pines tipo regleta	Sección de regleta macho de 10 pines	2	IN_SERVOS
Conector hembra de seguridad de 2 pines	Conector hembra de tornillo de 2 pines	11	F_C1-10, V_SERVOS
Conector hembra para cable de red	RJ 45	1	FINALES_CARRERA
Tarjeta de servos	BOARD 3	1	BRD 3

4.2.3. Board de potencia (Board 3). Esta etapa consta de los drivers usados para el manejo de las luces, teniendo en cuenta que las luces usadas en la valla son todas de 120 VAC se tomaron muchas medidas de seguridad, y la selección de los componentes de esta etapa fue una tarea muy dispendiosa.

Ya teniendo una idea de la función de la etapa, se realizó una caja negra cuya entrada es una señal TTL y su salida es una señal de 120 VAC, y con la garantía de poder lograr este comportamiento con frecuencias de activación de hasta 10 veces por segundo dependiendo de la programación de la valla y el efecto lumínico esperado.

La primera parte del diseño de esta etapa fue la selección del componente encargado de activar y desactivar las luces; después de un proceso de investigación y benchmarking, se llegó a la conclusión que los elementos mas usados para llevar a cabo esta tarea son los Relevos y los Triacs; ya la parte de acondicionamiento de la señal TTL se diseñó simplemente con una etapa de optoacoplador debido a su bajo costo tanto en tamaño como económico, entonces simplemente la señal proveniente del puerto paralelo que ya pasó por un driver 74

LS 244, llega a la etapa de potencia y se encuentra con un LED monitor que está conectado en serie con el LED interno del optoacoplador.

El LED monitor se usó en vez de la típica resistencia porque aparte de ocupar el mismo espacio y cumplir la misma función que es reducir el voltaje de +5VCC a +2.5 – 3VCC que requiere el LED del optoacoplador, sirve para visualizar la secuencia y de esta forma tener una realimentación visual del comportamiento de la etapa de potencia y descartar fácilmente elementos de la línea de control desde el PC hasta las luces en caso de fallas.

La selección del elemento encargado de la activación de las luces de 120 VAC se hizo mediante pruebas de 1 mes de funcionamiento continuo donde se implementaron 4 módulos de luces con relevos y 4 módulos de luces con triacs, cada uno con una secuencia donde las luces de 25 Watts se encendían y apagaban 10 veces por segundo durante los primeros 15 días y 0.2 veces por segundo durante los 15 días restantes de la prueba.

Al cabo del mes, los dos sistemas seguían funcionando en perfectas condiciones, entonces se llegó a la conclusión que la mejor opción era el triac porque tenía gran ventaja en precio y tamaño frente al relevo.

Como ya se había escogido el triac, se decidió seleccionar un optotriac para la activación del triac mediante la señal TTL, entonces se usó la pareja del optotriac MOC 3010 y triac 5645, la cual cuenta con un gran ahorro de espacio y a bajo precio.

Además, las luces que se van a usar en la valla no pasan de 15 Watts cada uno, por lo que se facilita la selección de los componentes para dicha etapa, sobre todo del triac que a parte de su buen desempeño nunca se calentó durante el mes de funcionamiento continuo con bombillos de 25 Watts.

Después de seleccionar los componentes para la etapa de potencia, se diseñó una tarjeta para alojar dichos componentes, teniendo en cuenta que son 8 líneas de control disponibles para conectar las luces, y como la salida en la tarjeta de comunicaciones es por medio de un conector tipo Red de 8 pines, simplemente se usó el mismo tipo de conector, de manera que se conectan las dos tarjetas por medio de un cable de Red común y corriente.

Esta tarjeta también cuenta con un conector hembra de 2 pines con tornillos de seguridad para la alimentación de 120 VAC, y un conector de tierra (GND) ya que las señales de los 8 pines del conector tipo Red son señales lógicas puras y no

cuentan con otra referencia que no sea la misma tierra de la fuente en la tarjeta de comunicaciones.

Figura 24. Board 3

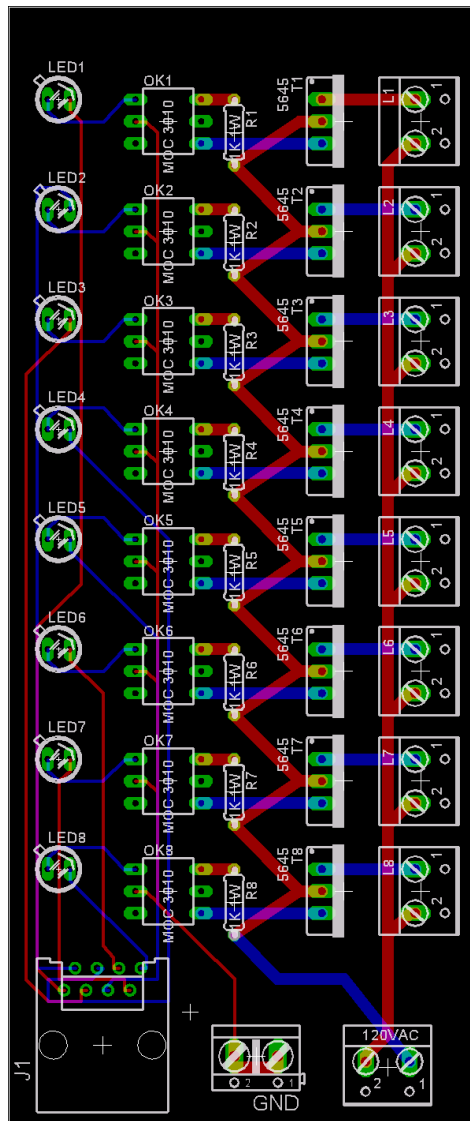


Figura 25. Diagrama esquemático de la Board 3

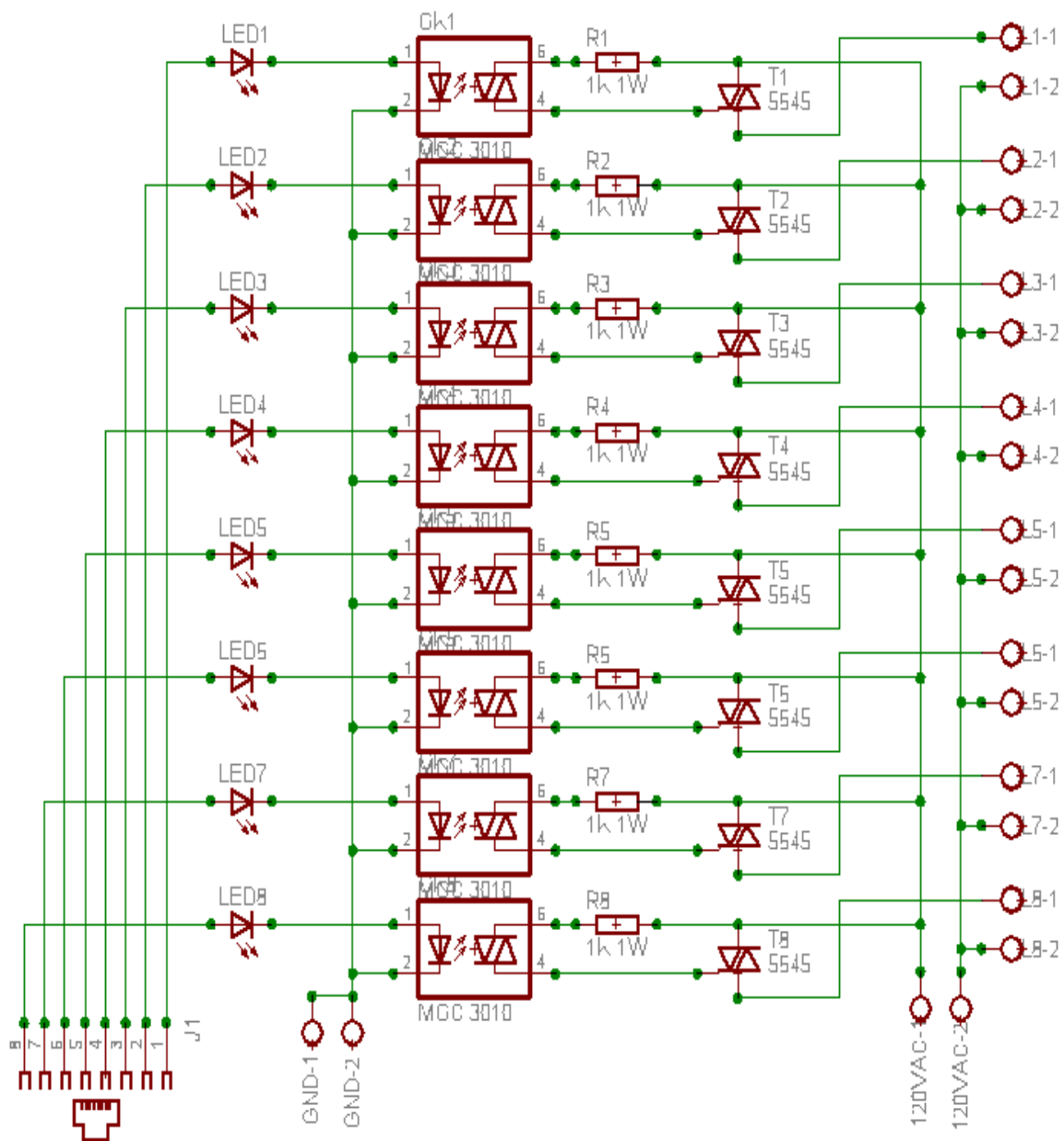


Tabla 8. Tabla de especificaciones de la Board 3

COMPONENTE	REFERENCIA	CANTIDAD	NOMBRE
Led	333 HD GREEN	8	LED1-8
Optotriac	MOC 3010	8	OK1-8
Resistencia	1k Ω , 1W	8	R1-8
Triac	NTE 5645	8	T1-8
Conector hembra de seguridad de 2 pines	Conector hembra de tornillo de 2 pines	10	L1-8, GND, 120 VAC
Conector hembra para cable de red	RJ 45	1	J1
Tarjeta de potencia	BOARD 2	1	BRD 2

5. DISEÑO Y ARQUITECTURA

5.1. Diagrama de clases.

En las figuras 26 y 27 se muestra el diagrama de clases desarrollado para el sistema. En la figura 26, se ha desglosado el sistema de control de reproducción, mientras que la figura 27 muestra todo el sistema de control, en ambos diagramas, la clase Frankie, es la clase principal y la que controla el flujo y los tiempos del programa.

Figura 26. Software, sistema de control de reproducción

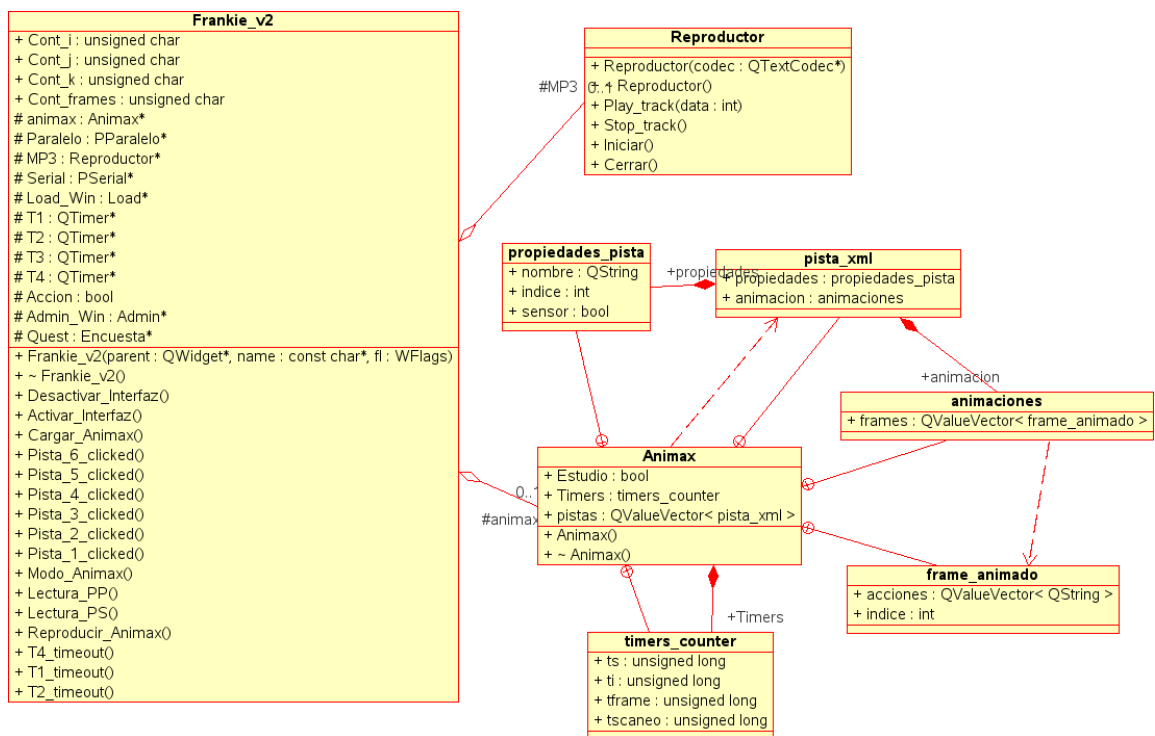
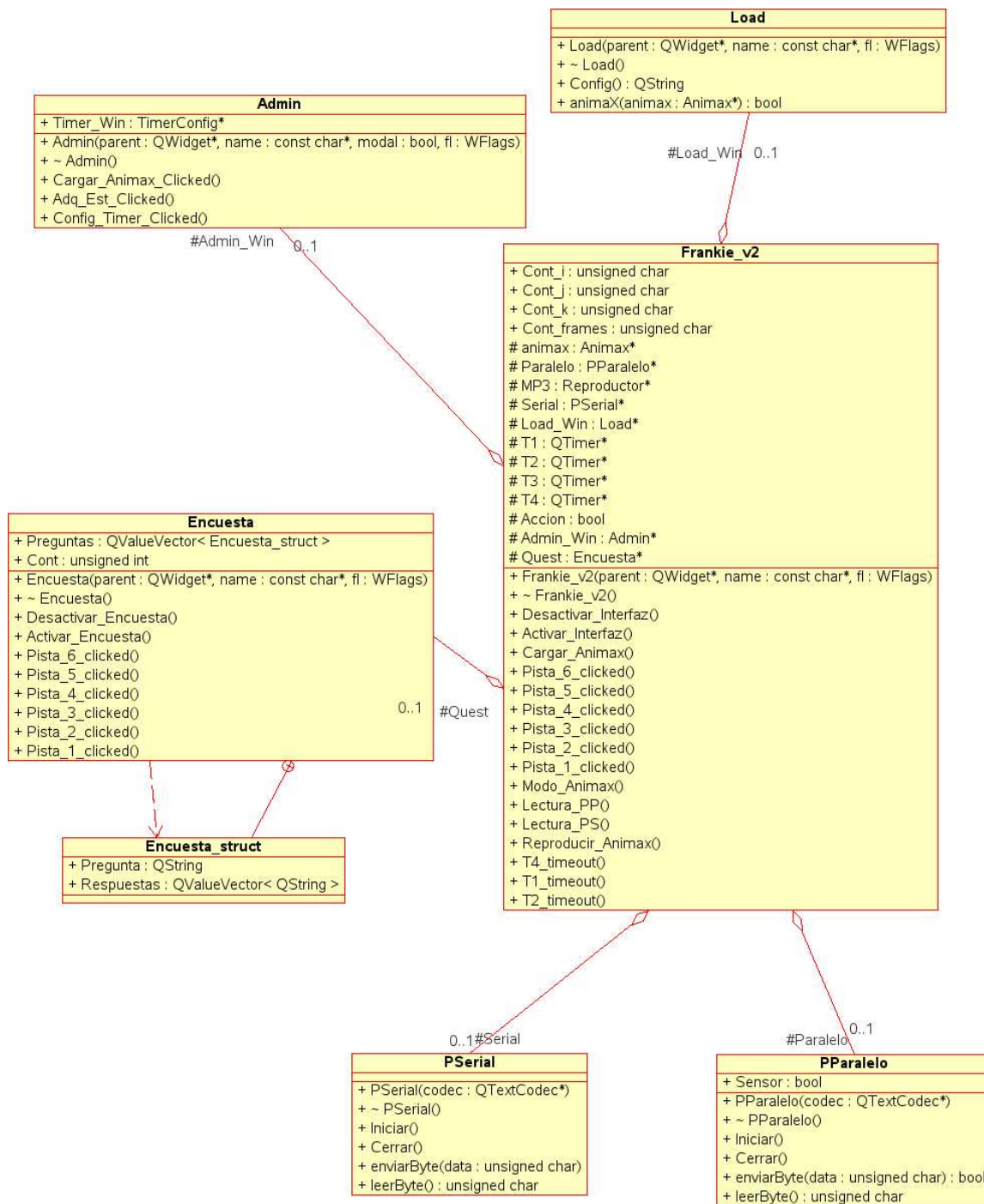


Figura 27. Software, sistema de control general



6. CONCLUSIONES

- El equipo de trabajo desarrolló una metodología ideal para llevar a cabo los objetivos del proyecto basándose en una plataforma estructurada partiendo desde los requerimientos de la empresa, y llegando hasta el desarrollo de las actividades a un nivel industrial aplicando de esta forma las habilidades desarrolladas durante el transcurso de la carrera.
- Las innovaciones tecnológicas y la introducción de la Mecatrónica en mercados en desarrollo en Colombia, como por ejemplo el de la publicidad, generan buenos nichos laborales para los ingenieros Mecatrónicos, creando grandes oportunidades desde el punto de vista empresarial y con la posibilidad de patentar mecanismos o productos novedosos.
- La integración de elementos de Software y Hardware desarrollados para un producto desde sus requerimientos, garantizan que se pueda documentar todo el desarrollo de las etapas del producto y su posible rediseño en cualquier aspecto ya que la relación Software – Hardware se ha desarrollado de una manera cooperativa y partiendo desde las mismas necesidades.
- El uso de herramientas de software para el diseño mecánico y electrónico le brindan al diseñador la posibilidad de mostrar al cliente simulaciones y animaciones que ayudan a visualizar el desarrollo del producto y posibles cambios en el diseño antes de invertir sumas de dinero sin que el cliente haya quedado totalmente satisfecho con el diseño mecánico, electrónico y de software, logrando un mejor entendimiento por parte del cliente y una herramienta indispensable a la hora de explicar cualquier aspecto de diseño.
- El resultado de aplicar la metodología de Diseño Mecatrónico I y II, así como también de Desarrollo de Productos, es muy positivo si se realiza el trabajo a conciencia; con estas metodologías se puede saber en cualquier momento en que punto se encuentra determinado proyecto en el que se este trabajando, tener indicadores y medir variables que nos lleven a predecir situaciones lo que nos prepararía para afrontar contratiempos y ejecutar planes alternativos sin detener los procesos de diseño, desarrollo e implementación.
- Los Ingenieros Mecatrónicos han demostrado que pueden encargarse tanto de la parte gerencial como también de la parte operacional de un proyecto, su

formación es muy completa y cuentan con herramientas tecnológicas tanto de software como de hardware, y también muchas bases de diseño mecánico, lo que les permite ser muy competitivos frente a ingenieros de otras ramas, destacándose en el diseño de soluciones a problemas de ingeniería, con liderazgo, tecnología, creatividad y productividad.

BIBLIOGRAFÍA

GONZALES MALDONADO, Fabián Andrés; RIASCOS MENA, Diego. Diseño y Construcción de Valla Robótica para Publicidad, Módulo 1. Trabajo de grado Ingeniero Mecatrónico. Santiago de Cali: Universidad Autónoma de Occidente. Facultad de Ingenierías, 2006. 78 p..

NORTON, Robert. Diseño de Máquinas. 4 ed. México: Prentice Hall, 1999. 1080 p.

ULRICH, Kart; EPPINDER, Steven. Product Design and Development. 2 ed. México: McGraw Hill, 2000. 358 p.

Valla Robótica para Publicidad ANIMAK [en línea]. Bogotá Colombia: KAFKA Ltda, 2006. [Consultado 12 de marzo de 2007]. Disponible en internet:
<http://kafkatek.com/proyectos/animak>

ANEXOS

Anexo A. Documentación de Linux

¿Que es Linux?

El término "Linux" estrictamente se refiere al núcleo Linux, pero es comúnmente utilizado para describir al sistema operativo tipo Unix (que implementa el estándar POSIX), que utiliza primordialmente filosofía y metodologías libres (también conocido como GNU/Linux) y que está formado mediante la combinación del núcleo Linux con las bibliotecas y herramientas del proyecto GNU y de muchos otros proyectos/grupos de software (libre o no libre). Actualmente, una parte considerable de las aplicaciones que conforman un sistema GNU/Linux son de origen distinto: GNU, KDE, OpenOffice.org, Mozilla, Apache, MySQL, PostgreSQL, Ruby, Perl, Python, entre otros. El proyecto GNU posee desde hace muchos años su propio núcleo en desarrollo, llamado Hurd. El núcleo Linux se distribuye bajo los términos de la licencia GNU GPL en su versión 2.

La expresión "Linux" también es utilizada para referirse a las distribuciones GNU/Linux, colecciones de software que suelen contener grandes cantidades de paquetes además del núcleo. El software que suelen incluir consta de una enorme variedad de aplicaciones, como: entornos gráficos, suites ofimáticas, servidores web, servidores de correo, servidores FTP, etcétera. Coloquialmente se aplica el término "Linux" a éstas. Algunas personas opinan que es incorrecto denominarlas distribuciones Linux, y proponen llamarlas sistema GNU/Linux. Otras personas opinan que los programas incluidos proceden de fuentes tan variadas que proponen simplificarlo denominándolo simplemente a "Linux".

La marca Linux (Número de serie: 1916230) pertenece a Linus Torvalds y se define como "un sistema operativo para computadoras que facilita su uso y operación".

Mientras que GNU/Linux pretende que este sistema se mantenga como Software Libre. La popularidad del sistema se ha incrementado en el mercado de desde servidores, hasta Teléfonos portátiles. Su gran flexibilidad ha permitido que sea utilizado en un rango muy amplio de sistemas de cómputo y arquitecturas: computadoras personales, supercomputadoras, dispositivos portátiles, etc. Los sistemas GNU/Linux funcionan sobre más de 20 plataformas diferentes de hardware; entre ellas las más comunes son las de los sistemas compatibles con

PCs x86 y x86-64, computadoras Macintosh, PowerPC, Sparc, MIPS y la nueva consola de juegos PS3.

¿Que es el kernel/núcleo?

El kernel o núcleo de Linux se puede definir como el corazón de este sistema operativo. Es el encargado de que el software y el hardware de tu ordenador puedan trabajar juntos.

Las funciones más importantes del mismo, aunque no las únicas, son:

- Administración de la memoria para todos los programas y procesos en ejecución.
- Administración del tiempo de procesador que los programas y procesos en ejecución utilizan.
- Es el encargado de que podamos acceder a los periféricos/elementos de nuestro ordenador de una manera cómoda.

Características

- Multitarea: La palabra multitarea describe la habilidad de ejecutar varios programas al mismo tiempo. LINUX utiliza la llamada multitarea preeventiva, la cual asegura que todos los programas que se están utilizando en un momento dado serán ejecutados, siendo el sistema operativo el encargado de ceder tiempo de microprocesador a cada programa.
- Multiusuario: Muchos usuarios usando la misma máquina al mismo tiempo.
- Multiplataforma: Las plataformas en las que en un principio se puede utilizar Linux son 386-, 486-, Pentium, Pentium Pro, Pentium II, Amiga y Atari, también existen versiones para su utilización en otras plataformas, como amd64, Alpha, ARM, MIPS, PowerPC y SPARC.
- Multiprocesador: Soporte para sistemas con más de un procesador está disponible para Intel, AMD y SPARC.
- Funciona en modo protegido 386.

- Protección de la memoria entre procesos, de manera que uno de ellos no pueda colgar el sistema.
- Carga de ejecutables por demanda: Linux sólo lee del disco aquellas partes de un programa que están siendo usadas actualmente.
- Política de copia en escritura para la compartición de páginas entre ejecutables: esto significa que varios procesos pueden usar la misma zona de memoria para ejecutarse. Cuando alguno intenta escribir en esa memoria, la página (4Kb de memoria) se copia a otro lugar. Esta política de copia en escritura tiene dos beneficios: aumenta la velocidad y reduce el uso de memoria.
- Memoria virtual usando paginación (sin intercambio de procesos completos) a disco: A una partición en el sistema de archivos, con la posibilidad de añadir más áreas de intercambio sobre la marcha.
- La memoria se gestiona como un recurso unificado para los programas de usuario y para el caché de disco, de tal forma que toda la memoria libre puede ser usada para caché y ésta puede a su vez ser reducida cuando se ejecuten grandes programas.
- Librerías compartidas de carga dinámica (DLL's) y librerías estáticas.
- Se realizan volcados de estado (core dumps) para posibilitar los análisis post-mortem, permitiendo el uso de depuradores sobre los programas no sólo en ejecución sino también tras abortar éstos por cualquier motivo.
- Compatible con POSIX, System V y BSD a nivel fuente.
- Emulación de iBCS2, casi completamente compatible con SCO, SVR3 y SVR4 a nivel binario.
- Todo el código fuente está disponible, incluyendo el núcleo completo y todos los drivers, las herramientas de desarrollo y todos los programas de usuario; además todo ello se puede distribuir libremente. Hay algunos programas comerciales que están siendo ofrecidos para Linux actualmente sin código fuente, pero todo lo que ha sido gratuito sigue siendo gratuito.
- Control de tareas POSIX.
- Pseudo-terminales (pty's).
- Emulación de 387 en el núcleo, de tal forma que los programas no tengan que hacer su propia emulación matemática. Cualquier máquina que ejecute Linux parecerá dotada de coprocesador matemático. Por supuesto, si el ordenador ya

tiene una FPU (unidad de coma flotante), esta será usada en lugar de la emulación, pudiendo incluso compilar tu propio kernel sin la emulación matemática y conseguir un pequeño ahorro de memoria.

- Soporte para muchos teclados nacionales o adaptados y es bastante fácil añadir nuevos dinámicamente.
- Consolas virtuales múltiples: varias sesiones de login a través de la consola entre las que se puede cambiar con las combinaciones adecuadas de teclas (totalmente independiente del hardware de video). Se crean dinámicamente y puedes tener hasta 64.
- Soporte para varios sistemas de archivo comunes, incluyendo minix-1, Xenix y todos los sistemas de archivo típicos de System V, y tiene un avanzado sistema de archivos propio con una capacidad de hasta 4 Tb y nombres de archivos de hasta 255 caracteres de longitud.
- Acceso transparente a particiones MS-DOS (o a particiones OS/2 FAT) mediante un sistema de archivos especial: no es necesario ningún comando especial para usar la partición MS-DOS, esta parece un sistema de archivos normal de Unix (excepto por algunas restricciones en los nombres de archivo, permisos, y esas cosas). Las particiones comprimidas de MS-DOS 6 no son accesibles en este momento, y no se espera que lo sean en el futuro. El soporte para VFAT, FAT32 (WNT, Windows 95/98) se encuentra soportado desde la version 2.0 del nucleo y el NTFS de WNT desde la version 2.2 (Este ultimo solo en modo lectura).
- Soporte en sólo lectura de HPFS-2 del OS/2 2.1
- Sistema de archivos de CD-ROM que lee todos los formatos estándar de CD-ROM.
- TCP/IP, incluyendo ssh, ftp, telnet, NFS, etc.
- Appletalk.
- Software cliente y servidor Netware.
- Lan Manager / Windows Native (SMB), software cliente y servidor.
- Diversos protocolos de red incluidos en el kernel: TCP, IPv4, IPv6, AX.25, X.25, IPX, DDP, Netrom, etc.

Anexo B. Cronograma proyecto ANIMAK

CRONOGRAMA PROYECTO ANIMAK 2005-2006																		
ETAPAS DE DISEÑO ESTRUCTURADO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
IDENTIFICACIÓN DE NECESIDADES																		
Identificar requerimientos del cliente con todo el equipo KAFKA																		
ESPECIFICACIONES DEL PRODUCTO																		
Definición de métricas																		
Benchmarking con medios afines																		
Establecimiento de valores marginalmente aceptables																		
GENERACION Y SELECCION DE CONCEPTOS																		
Búsqueda externa																		
Búsqueda interna																		
Exploración sistemática																		
Selección de conceptos mediante un método específico																		
PRUEBA DE CONCEPTOS																		
Realización de prototipos y pruebas																		
DISEÑO A NIVEL DE SISTEMA																		
ARQUITECTURA DEL PRODUCTO																		
Establecimiento de la arquitectura																		
DISEÑO DETALLADO																		
DISEÑO INDUSTRIAL																		
Identificar requerimientos del cliente con todo el equipo KAFKA																		
Refinamiento preliminar																		
Refinamiento final y selección final de conceptos																		
DISEÑO PARA MANUFACTURA																		
Diseño detallado																		
Estimación de costos de manufactura																		
Reducción de costos de componentes																		
Reducción de costos de ensamble																		
Reducción de costos de producción																		
Consideraciones del impacto del DPM en otros factores																		
PRUEBAS Y REFINAMIENTO																		
PROTOTIPADO																		
Definición de prototipos																		
Establecimiento del nivel de aproximación de los prototipos																		
Construcción de prototipos																		
CONSTRUCCIÓN																		
INSTALACIÓN VALLA PILOTO																		